

THÈSES

présentées

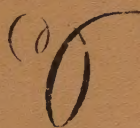
à la Faculté des Sciences de l'Université de Nantes

pour obtenir

le GRADE de DOCTEUR - INGENIEUR

par

Michel BELHACHE



1^{ère} THESE : Arithmétique pour Nombres Réels et Complexes
exprimés en Virgule Flottante FLAIR Multiple Précision

2^{ème} THESE : *Propositions données par la Faculté.*

Soutenues le 26 Octobre 1964 devant la Commission d'Examen

M. BRILLOUET Président

M. PIRONNEAU
M. PAUC } Examinateurs

THÈSES

présentées

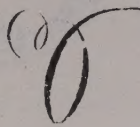
à la Faculté des Sciences de l'Université de Nantes

pour obtenir

le GRADE de DOCTEUR - INGENIEUR

par

Michel BELHACHE



1^{ère} THESE : Arithmétique pour Nombres Réels et Complexes
exprimés en Virgule Flottante **FLAIR** Multiple Précision

2^{ème} THESE : *Propositions données par la Faculté.*

Soutenues le 26 Octobre 1964 devant la Commission d'Examen

M. BRILLOUET Président

M. PIRONNEAU
M. PAUC } Examinateurs

UNIVERSITE de NANTES

FACULTE des SCIENCES

<i>Doyen</i>	M. BRILLOUET
<i>Assesseur</i>	M. PIRONNEAU

<i>Professeurs Titulaires</i>	MM. BRILLOUET	Mathématiques
	PAUC	"
	PETRESCU	"
	RAFFIN	"
	PIRONNEAU	Mécanique
	RAVILLY	"
	BOULLOUD	Physique
	COLLET	"
	DUNOYER	Chimie
	GRAFF	"
	VAUGIEN	Biologie Animale
	GARRIGUES	Biologie Végétale
	Mlle DEBRAUX	" "
	M. GOURINARD	Géologie

<i>Professeurs et Maîtres de Conférences</i>	MM. MOLINARO	Mathématiques
	VAILLANT	"
	VIVIER	"
	BALQUET	Mécanique
	AVIGNON	Physique
	GOUREAUX	"
	LASKAR	"
	LERMUSIAUX	"
	MEZENCEV	"
	MINN	"
	MOUTON	"
	JOLIVET	Chimie
	GOBIN	"
	QUINIOU	"
	ROUXEL	"
	WOJTKOWIAK	"
	PIERI	Biologie Animale
	SELLIER	" "
	DUPONT	Biologie Végétale
	OTTMANN	Géologie
	FORESTIER	"

Secrétaire

M. PAILLET

La présente étude a été effectuée au Centre de Calcul Scientifique de l'Ecole Nationale Supérieure de Mécanique de Nantes et menée à bien sous la direction de Monsieur le Professeur BRILLOUET, Doyen de la Faculté des Sciences de Nantes, Directeur du Centre de Calcul Scientifique.

Je tiens à exprimer toute ma gratitude à Monsieur le Professeur BRILLOUET pour l'intérêt permanent qu'il a manifesté vis à vis de mon travail, pour les conseils qu'il a bien voulu m'accorder.

Ma reconnaissance s'adresse également à Monsieur le Professeur PAUC et à Monsieur le Professeur LE SQUIN ainsi qu'à Messieurs les Professeurs qui ont bien voulu me faire l'honneur de composer le Jury.

Je remercie vivement tous mes camarades du Centre de Calcul de l'E.N.S.M. ainsi que les personnes qui ont participé à la bonne présentation et à l'impression de ce fascicule.

Jeg tilegner min kone og vår lille Anne dette arbeidet som bevis på min kjoerlighet

INTRODUCTION

.....

L'emploi de la virgule flottante pour la représentation et le traitement des nombres sur Ordinateur I B M 650 permet de résoudre les problèmes dans lesquels le cadrage du résultat est à priori inconnu et en particulier, la plupart des problèmes scientifiques.

Mais dans cette représentation, puisque 2 positions décimales du Mot 650 qui reçoit le nombre traité doivent être réservées à l'exposant, il ne reste plus que 8 positions pour les chiffres significatifs. Et si, pour la majorité des problèmes scientifiques et techniques simples, ce nombre reste suffisant, pour certains autres problèmes, il peut se révéler trop petit.

Il y a d'abord le cas évident où l'on désire connaître les résultats avec un grand nombre de chiffres significatifs : par exemple, établissement d'une table de fonction avec 12 ou 15 décimales. Mais c'est également le cas de certains problèmes mathématiques initialement "mal conditionnés" ou bien exigeant un gros volume de calculs, pour lesquels, si l'on désire obtenir les résultats avec une précision déterminée, il devient nécessaire que les données initiales du problème soient connues avec une précision bien supérieure, toute imprécision sur ces dernières pouvant avoir, du fait de la propagation des erreurs au cours des calculs, des répercussions sur la solution du problème.

C'est pour cette raison qu'il est nécessaire de pouvoir utiliser une représentation des nombres réels et complexes avec plus de 8 chiffres significatifs. Le but du travail ici présenté est de pouvoir traiter ces nombres dans des conditions optimales sur l'Ordinateur 650, pour lequel la structure à Mot de longueur fixe (dix positions décimales et une position de signe) se trouve mal adaptée au problème posé.

Dans une première partie, nous avons défini, dans le cadre des caractéristiques particulières de l'Ordinateur I B M 650, une représentation des nombres réels, analogue à la représentation en virgule Flottante FLAIR, le nombre étant représenté par le couple exposant-mantisse, la mantisse comportant plus de huit chiffres significatifs. Nous avons alors cherché à adapter les algorithmes classiques des opérations arithmétiques élémentaires à cette nouvelle représentation des opérandes et du résultat d'une opération. Puis, toujours dans le cadre des caractéristiques particulières de l'Ordinateur I B M 650, nous avons précisé la structure d'une Arithmétique pour le traitement des nombres réels et d'une Arithmétique pour le traitement des nombres complexes.

Dans la seconde partie, nous nous sommes attachés à définir les conditions d'emploi des Arithmétiques réalisées, nous bornant à la double et à la triple précision pour les nombres réels, à la simple et à la double précision pour les nombres complexes. Deux exemples illustrent ces explications.

Enfin, nous plaçant du point de vue du programmeur et dans le but de lui simplifier le travail, nous avons introduit des nouveaux codes symboliques correspondant à chacune des opérations arithmétiques et

à chacun des transferts réalisables avec les différentes Arithmétiques programmées. Des règles de programmation précisent dans quelles conditions ces codes peuvent être utilisés. Un compilateur permet l'interprétation des instructions écrites avec ces codes et leur traduction en séquences directement interprétables par le programme d'assemblage PASO.

En annexe, on trouvera l'analyse détaillée des Arithmétiques et du Compilateur.

Tous nos essais ont été effectués sur l'Ordinateur I B M 650 standard du Centre de Calcul Scientifique de l'Ecole Nationale Supérieure de Mécanique et tous les programmes réalisés ont été contrôlés, d'une part, au moyen de jeux de tests appropriés et d'autre part, utilisés pour la résolution de différents problèmes.

LISTE DES PRINCIPALES NOTATIONS ET ABREVIATIONS

-oOo-

A	- Nombre réel, généralement le premier opérande de l'opération $A \star B = C$ - Partie réelle du premier opérande de l'opération entre nombres complexes. $(A, B) \star (A', B') = (C, D)$
A'	- Partie réelle du second opérande complexe.
A.Cx.D.P.	- Arithmétique Complexe Double Précision
A.Cx.SD.P.	- Arithmétique Complexe mixte, Simple et Double Précision
A.Cx.S.P.	- Arithmétique Complexe Simple Précision
A.D.P.	- Arithmétique réelle Double Précision
A I _j	- Mémoire du PACx contenant le Mot de rang j de la partie imaginaire d'un nombre Complexe.
A R _i	- Mémoire du PACx contenant le Mot de rang i de la partie réelle d'un nombre Complexe.
A.SD.P	- Arithmétique réelle mixte, Simple et Double Précision
A.T.P.	- Arithmétique réelle Triple Précision
A.V.F.	- Arithmétique Virgule Flottante, simple précision.
a	- Mantisse de A
a _i	- Tranche de rang i de a
α	- Exposant de A
B	- Nombre réel, généralement le second opérande de l'opération $A \star B = C$ - Partie imaginaire du premier opérande de l'opération entre nombres complexes $(A, B) \star (A', B') = (C, D)$
B'	- Partie imaginaire du second opérande complexe
b	- Mantisse de B
b _j	- Tranche de rang j de b

β	- Exposant de B
C	- Nombre réel ; résultat de l'opération $A * B = C$ - Partie réelle du résultat de l'opération entre nombres complexes $(A, B) * (A', B') = (C, D)$
C.O.P.	- Code Opération dans une instruction
• c	- Mantisse de C
• c_ℓ	- Tranche de rang ℓ de -c
γ	- Exposant de C
D	- Partie imaginaire du résultat de l'opération entre nombres complexes - Dans une division, le dividende .
D_i	- Partie de rang i du dividende
$D', D'' \dots$	- Distance entre Mots Consécutifs d'un même nombre
DI_j	- Mémoire du PDCx contenant le Mot j de la partie imaginaire du nombre complexe
DR_i	- Mémoire du PDCx contenant le Mot i de la partie réelle du nombre complexe.
d	- Diviseur dans une division - Mantisse de la partie imaginaire D
d_i	- Partie de rang i du diviseur - Tranche de rang i de la mantisse d .
oo du	- Différence $ \beta - \alpha $, d et u étant respectivement les chiffres des dizaines et des unités de cette différence .
Δ	- Distance entre Mots de même rang de la partie réelle et de la partie imaginaire d'un même nombre complexe
δ	- Exposant de la partie imaginaire D
E , e	- Exposants d'un nombre réel,
E_0	- Exposant de base

$F, F' \dots$	- Adresses des différents Mots d'un nombre réel ou de la partie réelle d'un nombre Complexe
F	- Dans une intruction, représente l'adresse facteur
$G, G' \dots$	- Adresses des différents Mots de la partie imaginaire d'un nombre complexe
H_i	- Mémoire de rang i du P D
I	- Dans une instruction, adresse de l'instruction suivante. Egalement adressede l'instruction de sortie de l'Arithmétique
i	- Indice variable
j	- "
K_i	- Mémoire de rang i du P A
L	- Terme correctif utilisé en division
ℓ	- Indice variable
ℓ_i	- Tranche de rang i du terme correctif L
M	- Mantisse d'un nombre réel
m_i	- Tranche de rang i de la mantisse M
MA_i, MB_i, MC_i	- Mémoires de travail
N	- Nombre réel
n	- Degré de précision du nombre traité - Nombre de tranches constituant la mantisse - Numéro de carte
p	- Nombre de chiffres d'un exposant
$P.A$	- Pseudo-Accumulateur pour Arithmétique réelle
$P.A.Cx$	- " " complexe pour Arithmétique complexe
$P.D.$	- Pseudo-Distributeur pour Arithmétique réelle
$P.D.Cx$	- " " complexe pour Arithmétique complexe

Q	- Quotient d'une division
q	- Nombre des chiffres significatifs d'une mantisse
q_i	- Partie de rang i du quotient
R	- Ensemble des nombres réels
R_i	- Reste de rang i d'une division
r_i	- Somme partielle de rang i dans l'addition de deux mantisses
S.P.E.	- Sous-Programme d'Entrée
S.P.O.	- " d'opération
S.P.S.	- " de sortie
S.P.T.	- " de transfert
T	- Mémoire de travail
u_p	- Terme de rang p d'une suite
V.F.M.P.	- Virgule Flottante Multiple Précision
x	- Nombre de positions à décaler - Partie réelle du nombre complexe z
xx ou xxxx	- Dans une instruction quantité inconnue à priori
y	- Partie imaginaire du nombre complexe z
Z	- Nombre complexe $Z \equiv (A, B)$
z	- " complexe $z = x + iy$

CHAPITRE I

Nombres

Virgule Flottante Multiple Précision

Schéma d'une Arithmétique pour 650

SECTION 1.1

REPRESENTATION DES NOMBRES V F M P

OPERATIONS ARITHMETIQUES

1.11 - REPRESENTATION DES NOMBRES REELS EN VIRGULE FLOTTANTE - MULTIPLE PRECISION

1.111 - INTRODUCTION

Tout nombre N de l'ensemble R des réels peut être représenté sous la forme du produit :

$$N = M \times 10^e \quad (1)$$

dans lequel M est un nombre réel formé des chiffres de N affecté de son signe et 10^e est une puissance de 10 exprimant l'ordre de grandeur de N . Par exemple, le nombre $-123,45$ pourra s'écrire sous la forme :

$$(-1,2345) \times 10^2.$$

A tout nombre N il est donc possible d'associer au moins un couple (e, M) qui le représente.

M et par suite e seront uniques si, par convention nous imposons à M une forme décimale définie par la double inégalité :

$$0,1 \leq M < 1 \quad (2)$$

Par suite, à tout nombre N de l'ensemble R , il est possible de faire correspondre un couple (e, M) qui le représente biunivoquement :

$$N \in R \iff (e, M) \text{ tel que :}$$

$$N = M \times 10^e \text{ avec } 0,1 \leq M < 1.$$

e s'appelle l'exposant et M , la mantisse du nombre N .

En reprenant l'exemple ci-dessus, nous aurons la relation : $-123,45 \iff [3, -0,12345]$.

1.112 - CONDITIONS D'UTILISATION

Avant de définir plus précisément cette représentation des nombres réels, il importe de préciser dans quelles conditions ces nombres seront traités.

Les nombres réels représentés sous cette forme seront traités sur I B M 650 Standard, dont le Mot de longueur fixe renferme 10 positions décimales et une seule position de signe.

D'autre part, les nombres traités sous cette représentation devant être chargés en mémoire sur le tambour magnétique, leur forme devra être aussi condensée que possible et il n'est pas concevable qu'un exposant de 2 ou 3 chiffres immobilise à lui seul la totalité d'un Mot.

Il résulte de ces considérations que dans la représentation en Virgule Flottante Multiple Précision (en abrégé, VFMP), l'exposant devra être inclus à l'un des Mots qui contient une partie de la mantisse d'où il découle certaines adaptations que nous allons étudier pour l'exposant et la mantisse.

1.113 - REPRESENTATION DE L'EXPOSANT

L'exposant e défini par la relation (1) est un nombre entier, positif, négatif ou nul alors que M est également un nombre algébrique. Mais, puisque le Mot 650 ne possède qu'une seule position de signe et que par souci d'économie l'un des Mots 650 contenant les différentes parties de M devra porter à la fois l'exposant et une partie de cette mantisse, nous réserverons la position de signe pour y placer le signe de M et ferons en sorte que e soit représenté par un nombre essentiellement positif.

Par abus de langage on appellera encore EXPOSANT, le nombre entier E , positif, formé de p chiffres, exprimant l'ordre de grandeur de N et tel que :

$$E = e + E_0$$

E_0 étant l'exposant central choisi de sorte que :

$$0 < E < 10^p$$

on prendra :

$$E_0 = \frac{10^p}{2}$$

Les limites permises pour e sont donc $-\frac{10^p}{2}$ et $+\left(\frac{10^p}{2} - 1\right)$ ce qui restreint l'ordre de grandeur des nombres pouvant être traités avec cette représentation. Dans la réalisation pratique de cette Arithmétique il faudra tenir compte de ces limites et prévoir des sécurités lorsque E devient négatif ou supérieur à $(10^p - 1)$.

Avec cette représentation le nombre unité qui s'écrit $1 = 0,1 \times 10^1$ aura pour exposant

$$E(1) = E_0 + 1 = \frac{10^p}{2} + 1$$

Remarque

1. Par convention le nombre zéro a un exposant et une mantisse nuls.
2. Si de deux exposants E_i et E_j on choisit le plus grand, on posera :

$$\begin{aligned} \text{Sup}(E_i, E_j) &= E_i && \text{si } E_i > E_j \\ &= E_j && \text{si } E_i < E_j \\ &= E_i = E_j && \text{si } E_i = E_j \end{aligned}$$

1.114 - REPRESENTATION DE LA MANTISSE

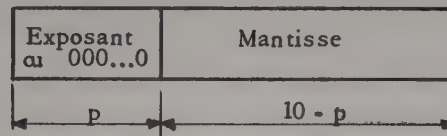
1 - Explicitation de M

Le nombre réel N se trouvant sous la forme $N = M \times 10^e$, la mantisse M est un nombre algébrique formé des chiffres de N , affecté de son signe et présentant une forme décimale : $0,1 \leq M < 1$

Puisque cette mantisse est traitée sur I B M 650, à Mot de longueur fixe et à 10 positions décimales, il sera nécessaire de répartir les chiffres de la mantisse en tranches de 10 chiffres occupant la totalité d'un Mot, sauf l'une des tranches, qui, devant également contenir l'exposant de N , ne portera que $(10 - p)$ chiffres de la mantisse, de sorte que la mantisse d'un nombre Virgule Flottante Multiple Précision à n Mots comportera $(10n - p)$ chiffres dont certains peuvent être nuls à droite.

Chaque tranche est affectée d'un indice i spécifiant son rang dans la suite des tranches qui constitue M . La tranche de rang 1 ne contient que les $(10 - p)$ premiers chiffres de la mantisse de la partie droite ; les p positions de gauche sont soit libres, soit réservées à l'exposant (voir Remarque).

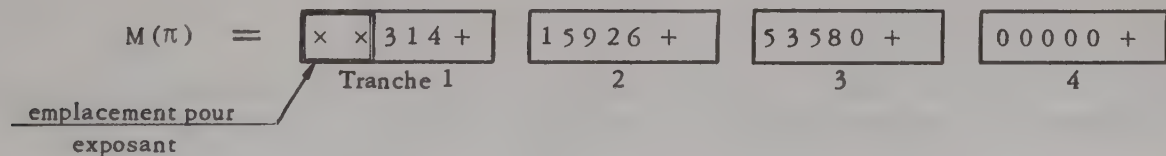
Tranche 1 :



Les 10 autres chiffres immédiatement à droite constituent la tranche 2 et ainsi de suite de 10 en 10 chiffres jusqu'à la $n^{\text{ième}}$ tranche formée des 10 derniers chiffres à l'extrême droite. Lorsque le nombre q des chiffres significatifs de N est inférieur au nombre $(10n - p)$ des positions disponibles, les positions à droite à $q^{\text{ième}}$ seront remplies de zéros.

Toutes les tranches sont affectées d'un même signe qui est celui de N .

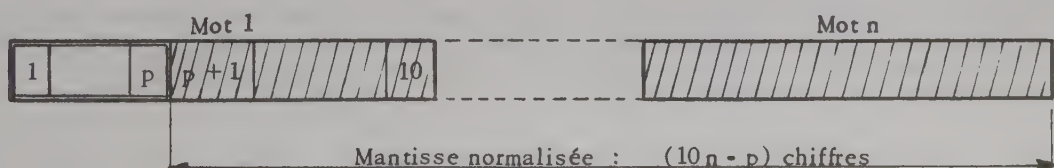
Par exemple, la représentation de la mantisse du nombre π supposé connu avec 12 chiffres significatifs dans un système de 4 tranches à 5 chiffres et pour un exposant de 2 chiffres sera :



2 - Mantisse normalisée

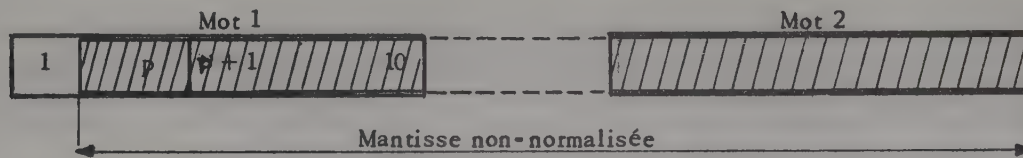
La mantisse, telle que nous venons de la définir :

- sous forme décimale
- le premier chiffre significatif occupant la $(p + 1)^{\text{ième}}$ position gauche du Mot 1, est dite "normalisée".



Associée à l'exposant E défini au paragraphe précédent, elle représente le nombre N en Virgule Flottante, Multiple Précision.

Par contre, si le premier significatif de la mantisse se trouve à droite ou à gauche de la $(p + 1)$ ème position du Mot 1, la mantisse est dite "non-normalisée".



Pour une telle mantisse qui ne vérifie plus la relation de définition $0,1 \leq M < 1$, l'exposant E devra être corrigé afin que, après "normalisation" - recadrage sur la $(p + 1)$ ème position gauche du Mot 1 - le couple corrigé (E^*, M^*) représente sous forme VFMP le nombre précédemment défini. Cette forme non-normalisée se présente à l'intérieur de l'Arithmétique alors que l'exposant et mantisse ont été séparés, soit parce que l'opération envisagée sur les mantisses nécessite leur décalage à droite ou à gauche de la $(p + 1)$ ème position du Mot 1, soit comme mantisse-résultat même si les mantisses des opérandes étaient initialement normalisées.

3 - Représentation de M

Si nous reprenons l'écriture décimale d'un nombre réel, nous savons que dans cette représentation le rang de chaque chiffre correspond à une puissance de 10, négative pour les chiffres à droite de la virgule, nulle pour le premier chiffre à gauche et positive à gauche de cette position. Par exemple, 12,345... peut s'écrire : $(1 \times 10^1) + (2 \times 10^0) + (3 \times 10^{-1}) + (4 \times 10^{-2}) + (5 \times 10^{-3}) + \dots$

Par analogie et puisque M se présente sous forme décimale, totalement à droite de la virgule, nous la représenterons sous la forme:

$$M = (m_1 \times 10^{-10}) + (m_2 \times 10^{-20}) + \dots + (m_n \times 10^{-10n})$$

où

$$M = \sum_{i=1}^n m_i \times 10^{-10i} \quad \text{avec } m_i < 10^{10}$$

chaque m_i étant la tranche de rang i formée des 10 chiffres significatifs de M correspondant, de sorte que au rang i correspond une puissance de 10 égale à $-10i$;

4 - Décalage à droite

De même décaler un nombre de x positions, à droite par rapport à une position repère, revient à diminuer l'ordre de grandeur de ce nombre et en fait à le multiplier par 10^{-x} . Le décalage d'une tranche m_i vers la droite de x positions (x entier positif) lui fait correspondre la tranche m'_i de structure interne identique mais telle que $m'_i = m_i \times 10^{-x}$.

Pour décaler une mantisse, il suffit de décaler chacune de ses tranches de la même quantité.

$$M \times 10^{-x} = \sum_{i=1}^n (m_i \times 10^{-10i}) \times 10^{-x} = \sum_{i=1}^n m_i \times 10^{-(10i+x)}$$

1.115 - EN RESUME

A tout nombre $N \in \mathbb{R}$, il est possible d'associer biunivoquement un couple (E, M) qui le représente en VFMP pour 650 et tel que, si on écrit :

$$N = M \times 10^E \quad \text{avec} \quad 0,1 \leq M < 1,$$

nous ayons les relations :

$$\text{Exposant } E = e + E_0 \quad \text{de } p \text{ chiffres, entier positif avec } E_0 = \frac{10P}{2}$$

Mantisse $M = \sum_{i=1}^n m_i \times 10^{-10i}$, les m_i étant des tranches de 10 chiffres de M de rang i , sauf la première qui, dans certains cas, n'en a que $10 - p$. La mantisse M est normalisée.

1.116 - REMARQUES

Un nombre N , représenté en VFMP peut apparaître sous 2 formes, suivant qu'il est traité à l'extérieur ou à l'intérieur de l'Arithmétique.

1 - A l'extérieur de l'Arithmétique, la mantisse doit être normalisée de manière que l'exposant puisse occuper les p positions gauches du Mot 1. N n'utilise alors que n Mots 650 ; cette forme est dite "n Mots". Elle est utilisée pour l'écriture et le traitement des nombres VFMP en dehors de l'Arithmétique.

2 - A l'intérieur de l'Arithmétique, mantisse et exposant sont traités séparément. L'exposant immobilise un Mot et la mantisse formée de $(10n - p)$ chiffres immobilise n Mots ($p < 10$). Cette forme est dite "(n + 1) Mots". Suivant la nature et le développement des opérations en cours, la mantisse peut être ou non normalisée; en fin d'opération, la mantisse-résultat est généralement non normalisée.

Le fait qu'un nombre VFMP se présente sous 2 formes différentes, nécessitera de prévoir :

- d'une part, à l'entrée dans l'Arithmétique, la conversion de la forme "n Mots" en forme "(n + 1) Mots".
- d'autre part, à la sortie de l'Arithmétique, la normalisation de la mantisse - résultat et le retour à la forme "n Mots".

1.12 - OPERATIONS SUR LES NOMBRES REELS REPRESENTES EN VIRGULE FLOTTANTE MULTIPLE

PRECISION

A partir des exposants et des mantisses des 2 opérandes représentés sous forme VFMP, nous expliciterons les relations qui définissent l'exposant et la mantisse du résultat de l'opération dans le cas des 4 opérations arithmétiques : addition, soustraction, multiplication et division.

Remarquons que les règles données ne supposent pas que les mantisses sont sous forme normalisée mais simplement qu'elles sont initialement cadrées par rapport à une position définie servant de repère tout au long de l'opération.

1.121 - ADDITION

1.121.1 - Représentation de l'addition

Pour additionner 2 nombres A et B de l'ensemble R, on place ces 2 nombres l'un au-dessous de l'autre de manière qu'à l'intérieur de chacun d'eux, les chiffres de même rang décimal (unité, dizaine...) soient dans une même colonne. Le résultat ou somme est un nombre C de R, dont l'ordre de grandeur est celui de l'opérande le plus grand et dont les chiffres significatifs sont obtenus par addition des chiffres de même ordre décimal des 2 opérandes, compte tenu des reports éventuels.

Règles d'addition pour nombres VFMP

En s'inspirant du schéma précédent, nous distinguerons 3 phases dans le processus de l'addition des nombres VFMP :

1 - Recherche de l'opérande le plus grand, c'est-à-dire de l'exposant le plus grand

2 - Alignement des chiffres significatifs, donc des mantisses des 2 opérandes :

Une différence x entre les 2 exposants correspond à un rapport des nombres égal à 10^{-x} , donc à un décalage relatif de x positions vers la droite de la mantisse du plus petit des opérandes.

3 - Addition des 2 mantisses alignées .

Donc, étant donné, les 2 opérandes $A = (\alpha, a)$ et $B = (\beta, b)$, l'addition $A + B = C$ définit un nombre $C = (\gamma, c)$ tel que :

$$\begin{aligned} \gamma &= \text{Sup} (\alpha, \beta) \\ C &= a \times 10^{-(\gamma-\alpha)} + b \times 10^{-(\gamma-\beta)} \end{aligned}$$

1.121.2 - Remarques

1 - Les minuscules grecques ($\alpha, \beta, \gamma \dots$) désignent les exposants et les minuscules latines ($a, b, c \dots$) les mantisses à n tranches de 10 chiffres des opérandes. Dans le cas de l'addition, la tranche 1 des mantisses est formée des $(10 - p)$ premiers chiffres de la mantisse précédés à gauche de p zéros. Moyennant cette remarque, la tranche 1 est traitée comme une tranche quelconque.

2 - γ étant l'une des 2 quantités α et β , l'une des deux différences $(\gamma - \alpha)$ et $(\gamma - \beta)$ est nulle ; la mantisse correspondante n'est donc pas décalée et sert de repère. L'autre mantisse est décalée vers la droite ; ses chiffres significatifs qui passeront au delà de la $(10 - n)$ ième position seront alors perdus puisque la mantisse du résultat est à n tranches de 10 chiffres. Afin de préserver la précision du résultat, la mantisse décalée devra être arrondie sur sa position la plus à droite.

3 - Rappelons la règle d'arrondi utilisée par l'I B M 650 : pour arrondir un nombre sur le chiffre de rang n , on ajoute 5 en valeur absolue au chiffre de rang $n + 1$.

1.121.3 - Opération se déduisant de l'addition

- Soustraction : La différence de 2 nombres A et B est un nombre C défini par l'addition de A et de l'opposé de B :

$$C = A + (-B)$$

d'où

$$\gamma = \text{Sup}(\alpha, \beta)$$

$$C = a \times 10^{-(\gamma - \alpha)} + (-b) \cdot 10^{-(\gamma - \beta)}$$

- Opération en valeur absolue : On définit également une addition et une soustraction de valeur absolue qui se ramènent à une addition ou à une soustraction, suivant le signe de l'opérande dont on considère la valeur absolue.

$$A + |B| = A + B \text{ si } B \text{ est positif}$$

$$\text{ou } A - B \text{ si } B \text{ est négatif}$$

$$A - |B| = A - B \text{ si } B \text{ est positif}$$

$$\text{ou } A + B \text{ si } B \text{ est négatif}$$

1.122 - MULTIPLICATION

Etant donné les 2 nombres réels $A = (\alpha, a)$ et $B = (\beta, b)$, la multiplication $A \times B = C$ définit un nombre réel $C = (\gamma, c)$ appelé produit des 2 nombres A et B ; ce produit est défini par l'algorithme classique des multiplications répétées du multiplicande avec décalage.

Comme les 3 nombres A, B et C peuvent s'écrire : $A = a \times 10^{\alpha'}$, $B = b \times 10^{\beta'}$ et $C = c \times 10^{\gamma'}$ nous avons les 2 relations :

$$c = a \times b$$

et

$$\gamma' = \alpha' + \beta'.$$

1.122.1 - Expression de l'exposant γ

Dans la représentation en VF, MP nous avons par définition :

$$\gamma = \gamma' + E_0$$

$$\alpha = \alpha' + E_0$$

$$\beta = \beta' + E_0$$

d'où l'expression de γ

$$\gamma = \alpha + \beta - E_0$$

1.122.2 - Expression de la mantisse c

Nous avons les relations de définition :

$$a = \sum_{i=1}^n a_i \times 10^{-10i}$$

$$b = \sum_{j=1}^n b_j \cdot 10^{-10j} \quad \text{et} \quad c = \sum_{\ell=1}^n c_{\ell} \cdot 10^{-10\ell}$$

les a_i , b_j et c_{ℓ} étant des tranches de 10 chiffres.

Dans le traitement des mantisses en multiplication, un décalage de la totalité de la mantisse vers la gauche transporte les zéros des p positions libres de la première tranche à la fin de la dernière tranche, à droite du dernier chiffre significatif. Les mantisses non-normalisées sont cadrées sur la 1ère position gauche du Mot 1,

Sur le schéma de l'expansion de multiplication, considérons un produit $a_i \times b_j$; a_i est une tranche de 10 chiffres de rang i et b_j une tranche de 10 chiffres de rang j . Le produit $a_i \times b_j$ est constitué de 20 chiffres que l'on partagera en 2 tranches :

1 "La gauche de $a_i \times b_j$ " = $G(a_i \times b_j)$ formée des 10 chiffres de gauche du produit $a_i \times b_j$, et

2 "La droite de $a_i \times b_j$ " = $D(a_i \times b_j)$ formée des 10 chiffres de droite du produit $a_i \times b_j$.

Le rang de $D(a_i \times b_j)$ est $(i+j)$ alors que celui de $G(a_i \times b_j)$ est $(i+j-1)$. Symboliquement, on pourra écrire :

$$(a_i \times 10^{-10i}) \times (b_j \times 10^{-10j}) = G(a_i \times b_j) \cdot 10^{-10(i+j-1)} + D(a_i \times b_j) \cdot 10^{-10(i+j)}$$

Par exemple nous aurons :

$$(a_2 \times 10^{-20}) \times (b_3 \times 10^{-30}) = G(a_2 \times b_3) 10^{-40} + D(a_2 \times b_3) \cdot 10^{-50}$$

Si maintenant nous considérons une tranche c_{ℓ} du produit, de rang ℓ , cette tranche sera définie par une double somme :

- la somme de toutes les tranches gauches G de rang ℓ , c'est-à-dire correspondant aux produits $a_i \times b_j$ tels que $i+j-1 = \ell$
- et la somme de toutes les tranches droites D de rang ℓ , c'est-à-dire correspondant aux produits $a_i \times b_j$ tels que $i+j = \ell$.

Cela peut donc s'écrire :

$$c_{\ell} = \sum_{i,j=1}^n G(a_i \times b_j)_{i+j=\ell+1} + \sum_{i,j=1}^n D(a_i \times b_j)_{i+j=\ell}$$

Et par exemple, en se reportant au schéma de l'expansion de multiplication, on retrouve :

$$\begin{aligned} c_4 &= \sum_{i,j=1}^4 G(a_i \times b_j)_{i+j=5} + \sum_{i,j=1}^4 D(a_i \times b_j)_{i+j=4} \\ &= G(a_1 \times b_4) + G(a_2 \times b_3) + G(a_3 \times b_2) + G(a_4 \times b_1) + D(a_1 \times b_3) \\ &\quad + D(a_2 \times b_2) + D(a_3 \times b_1). \end{aligned}$$

Puisque les mantisses a et b sont à n tranches, leur produit est à $2n$ tranches dont les n premières, les plus à gauche, constituent la mantisse qui est également à n tranches. On pourra donc écrire :

$$c = \sum_{\ell=1}^n \left[\sum_{i,j=1}^n \left[G(a_i \times b_j)_{i+j=\ell+1} + D(a_i \times b_j)_{i+j=\ell} \right] \times 10^{-10\ell} \right]$$

Expansion de Multiplication

1	2	3	4	5	6	7	8
a_1	a_2	a_3	a_4				
b_1	b_2	b_3	b_4				
$a_1 \times b_1$							
	$a_1 \times b_2$						
		$a_1 \times b_3$					
			$a_1 \times b_4$				
	$a_2 \times b_1$						
		$a_2 \times b_2$					
			$a_2 \times b_3$				
				$a_2 \times b_4$			
		$a_3 \times b_1$					
			$a_3 \times b_2$				
				$a_3 \times b_3$			
					$a_3 \times b_4$		
			$a_4 \times b_1$				
				$a_4 \times b_2$			
					$a_4 \times b_3$		
						$a_4 \times b_4$	
c_1	c_2	c_3	c_4				

Remarque

Par souci de précision on calcule la tranche c_{n+1} dont on ne retiendra que le report d'arrondi sur la tranche c_n .

1.123 • DIVISION

Etant donné les 2 nombres réels $A = (\alpha, a)$ et $B = (\beta, b)$, la division de A par B :

$$A : B = C$$

définit un nombre réel $C = (\gamma, c)$ appelé quotient des 2 nombres A et B et défini par l'algorithme manuel de la division (construction du quotient chiffre par chiffre).

Comme les 3 nombres A , B et C peuvent s'écrire $A = a \times 10^{\alpha'}$ $B = b \times 10^{\beta'}$ et $C = c \times 10^{\gamma'}$, nous avons les relations :

$$c = a : b$$

et

$$\gamma' = \alpha' - \beta'$$

1.123.1 • Expression de l'exposant γ

Dans la représentation en VFMP, nous avons par définition :

$$\gamma = \gamma' + E_0$$

$$\alpha = \alpha' + E_0$$

$$\beta = \beta' + E_0$$

d'où une expression de γ

$$\gamma = \alpha - \beta + E_0$$

1.123.2 • Calcul de c par utilisation de l'algorithme manuel

1 • Méthode de la division manuelle

Le quotient est construit chiffre par chiffre. Au départ, on considère le premier chiffre d_1 du diviseur d et la partie correspondante D_1 du dividende D ; D_1 est formé du premier ou des 2 premiers chiffres de D . Le quotient q_1 de D_1 par d_1 est une première approximation du premier chiffre du quotient de D par d . Si q_1 est correct, le reste

$$R_1 = D - d \cdot q_1$$

devra vérifier la double inégalité :

$$0 \leq R_1 < d$$

Si cette double inégalité est vérifiée on poursuit la division par la recherche du second chiffre du quotient. Une première approximation en sera donnée par le quotient de la tranche suivante D_2 du dividende (formée de R_1 , s'il n'est pas nul, et du chiffre à droite de D_1 dans D) par d_1 . Et de même pour que ce quotient q_2 soit correct, le reste

$$R_2 = D - dq_2 \text{ devra vérifier } 0 \leq R_2 < d.$$

Et ainsi de suite, on construit le quotient de D par d de chiffre en chiffre, à chacun de ces chiffres le reste correspondant devant être positif ou nul mais inférieur au diviseur d .

Dans le cas où un reste ne vérifierait pas cette double inégalité, nous devrions modifier convenablement l'approximation du chiffre du quotient jusqu'à ce que le nouveau reste ainsi obtenu vérifie la double inégalité. Nous préciserons le sens de ces modifications au paragraphe suivant.

2 - Adaptation de l'algorithme manuel au calcul de c .

Les mantisses a , b et c étant constituées de n tranches de 10 chiffres (pour la tranche 1, voir au § 1.122.2) et puisque le 650 effectue les divisions dans des registres à 10 et 20 chiffres, on adaptera l'algorithme manuel de la division à la construction du quotient c , non pas chiffre par chiffre, mais tranche de 10 chiffres par tranche de 10 chiffres.

a) Au départ de la division.

Le diviseur est formé de la mantisse b et le dividende est formé de la totalité de la mantisse a , puisque a et b sont tous deux constitués de n tranches de 10 chiffres.

• 1ère phase : prévision

On effectue la division de a_1 par b_1 . Soit c_1 ce quotient ; il constitue une première approximation de la première partie du quotient.

• 2ème phase : vérification

En utilisant la multiplication, on effectue le produit de c_1 par b , que l'on compare à a . Il faut que le reste

$$R_1 = a - c_1 \times b \quad \text{vérifie la double inégalité} \quad 0 \leq R_1 < b$$

Eliminons immédiatement le cas où R_1 est nul ; la division est alors terminée, (voir Fin de division).

Considérons maintenant l'inégalité $R_1 < b$ et montrons qu'elle est toujours vérifiée.

A cet effet, coupons le dividende et le diviseur en 2 parties :

$$a = a_1 + a_2$$

$$b = b_1 + b_2$$

telles que a_1 soit directement divisible par b_1 dans les registres de l'Ordinateur IBM 650 (a_2 et b_2 sont constitués des autres chiffres des mantisses a et b) ; si c_1 est le quotient de a_1 par b_1 , on aura :

$$c_1 = \frac{a_1 - r}{b_1}$$

avec

$$0 \leq r < b_1$$

Le reste R_1 est défini par la relation

$$R_1 = (a_1 + a_2) - c_1(b_1 + b_2)$$

et nous devons vérifier que

$$R_1 < b_1 + b_2$$

$$\text{il vient alors } (a_1 + a_2) - \frac{(a_1 - r)}{b_1} (b_1 + b_2) < b_1 + b_2$$

$$b_1(a_1 + a_2) - (a_1 - r)(b_1 + b_2) < (b_1 + b_2)b_1$$

et après simplification

$$(a_2 b_1 - a_1 b_2) < (b_1 + b_2) (b_1 - r)$$

relation que nous devons vérifier.

Or, puisque r est inférieur à b_1 , le nombre $(b_1 - r)$ est un nombre strictement positif, de même rang que a_1 , donc $(b_1 - r) > a_2$ ou encore $(b_1 + b_2) (b_1 - r) > (b_1 + b_2) a_2$ avec $(b_1 + b_2) a_2 = b_1 a_2 + b_2 a_2 > b_1 a_2 - b_2 a_1$ donc :

$$(b_1 + b_2) (b_1 - r) > b_1 a_2 - b_2 a_1$$

ce qui vérifie bien la propriété de R_1 : $R_1 < b$

Le reste R_1 construit à partir de c_1 est donc toujours inférieur au diviseur b , c'est-à-dire que la première tranche du quotient de a par b est au plus égale à c_1 . Par contre R_1 peut être négatif.

Dernier cas : Si R_1 est négatif, cela signifie que la première approximation c_1 de la tranche du quotient est trop grande. Les corrections devront se faire dans le sens d'une augmentation du reste jusqu'à ce qu'il devienne positif ou nul. En effet, si nous écrivons cette augmentation sous la forme du produit $b \times h$, R_1 deviendra :

$$R_1 + b \cdot h,$$

donc :

$$\begin{aligned} R_1 + b \cdot h &= a - c_1 b + b h \geq 0 \\ &= a - b(c_1 - h) \end{aligned}$$

ce qui montre que si R_1 est négatif, il faudra diminuer c_1 .

Si le reste R_1 vérifie la double inégalité, la tranche c_1 est correcte et on poursuit la division par calcul de la tranche suivante c_2 .

• 3ème phase : correction

On diminue c_1 de une unité du dernier ordre à droite et on reprend la seconde phase avec ses mêmes conclusions pour $c'_1 = c_1 - 1$ et ainsi de suite jusqu'à ce que, de négatif, le reste devienne positif ou nul. A ce stade, on arrête la correction et le $c_1^{(n...n)}$ corrigé constitue la première partie du quotient.

b) Poursuite de la division

On constitue la partie suivante du dividende en adjoignant à droite de R_1 une tranche de 10 zéros. C'est cette nouvelle partie de dividende qui servira au calcul de c_2 et que l'on comparera au produit $c_2 \times b$ dans la reprise des 3 phases précédemment définies.

c) Fin de division

Dividende et diviseur ayant initialement même longueur, la division sera terminée lorsque pour une tranche c_i du quotient, le reste R_i est nul ; dans ce cas les tranches $c_{i+1}, \dots, c_n$ seront formées de zéros. Dans le cas général la division sera terminée lorsque le quotient c aura été déterminé avec n tranches.

1.123.3 - Calcul de c par développement en série

1. Expression de c

c est défini par le rapport de 2 mantisses a et b qui peuvent s'écrire :

$$a = \sum_{i=1}^n a_i \cdot 10^{-10 i} \quad b = \sum_{j=1}^n b_j \cdot 10^{-10 j}$$

où les a_i et b_j sont des tranches de 10 chiffres (voir § 1.122.2)

Nous avons :

$$b = b_1 \times 10^{-10} + \sum_{j=2}^n b_j \times 10^{-10 j}$$

ou encore :

$$b = b_1 \times 10^{-10} \times \left[1 + \frac{\sum_{j=2}^n b_j \times 10^{-10 j}}{b_1 \times 10^{-10}} \right]$$

de sorte que l'on pourra écrire :

$$c = \frac{a}{b_1 \times 10^{-10}} \times \frac{1}{1 + f}$$

$$\text{avec } f = \frac{\sum_{j=2}^n b_j \times 10^{-10 j}}{b_1 \times 10^{-10}}$$

et comme f est un nombre petit, inférieur à l'unité on pourra utiliser un développement en série et écrire :

$$\frac{1}{1 + f} = \sum_{m=0}^{\infty} (-1)^m \cdot f^m$$

c peut alors s'écrire $c = M \times L$ avec :

$$M = \frac{\sum_{i=1}^n a_i \times 10^{-10 i}}{b_1 \times 10^{-10}} \quad \text{et} \quad L = \sum_{m=0}^{\infty} (-1)^m f^m$$

Le calcul de M se fait dans les registres par une expansion de dividende, le diviseur gardant une longueur constante de 10 chiffres. M peut donc être calculé avec la précision désirée en développant plus ou moins la division. On se contente de calculer M avec n tranches de 10 chiffres et le cadrage initial de a et b_1 se fera de sorte que l'on ait :

$$0,1 \leq M < 1.$$

2 - Calcul de L

L est défini par une série alternée convergant très rapidement ; il faudra donc chercher à limiter son développement de sorte que le premier terme négligé soit inférieur à l'erreur maximale tolérée sur le résultat puisque M reste inférieur à 1.

La mantisse c étant formée de $10n - p$ chiffres, l'ordre décimal du dernier chiffre d'extrême droite est $10^{-(10n-p)}$. Ce chiffre étant arrondi, l'erreur maximale autorisée sur c est donc de une demi-unité de cet ordre, soit $\frac{1}{2} \times 10^{-(10n-p)}$.

Il faut donc que le premier terme négligé $(-f)^m$ soit en valeur absolue au plus égal à cette demi-unité.

$$|f^m| \leq \frac{1}{2} \times 10^{-(10n-p)}$$

Le maximum pour f sera obtenu lorsque dans la fraction qui le définit, le numérateur sera maximal et le dénominateur minimal ; soit donc pour :

$$b_1 = 100000000$$

$$\sum_{j=2}^n b_j \cdot 10^{-10j} = 999999999 \dots 9999$$

2^e tranche 10(n-1) - p neufs

ou en majorant

$$\sum_{j=2}^n b_j \cdot 10^{-10j} \quad \begin{array}{c} 00\,0000\,0001\,00\,000\,\dots \\ \text{1^e tranche} \quad \text{2^e tranche} \end{array}$$

de sorte que

$$f < \frac{00\,0000\,0001}{10\,0000\,0000} = 1 \times 10^{-9}$$

Il suffit que le premier terme négligé soit celui dont la puissance m est telle que :

$$(10^{-9})^m < \frac{1}{2} \times 10^{-(10n-p)},$$

d'où en passant au logarithme :

$$m > \frac{(10n-p)}{9} + \frac{\log 2}{9}$$

On pourra prendre $m = n + 1$, c'est-à-dire développer L jusqu'à la puissance n , pourvu que :

$$\begin{array}{ll} n \leq 11 & \text{pour } p = 2 \\ \text{ou } n \leq 12 & \text{pour } p = 3 \end{array}$$

On pose donc :

$$L = \sum_{m=0}^n (-1)^m f^m$$

Pour le calcul de cette somme on utilise le schéma de l'expansion de multiplication ainsi que l'addition.

Connaissant M et L , en utilisant l'expansion de multiplication on obtient c .

1.123.4 • Comparaisons

L'un et l'autre de ces algorithmes sont également valables, précis et susceptibles de se prêter à une programmation par boucle. Le premier seul permet de connaître le reste de la division; par contre pour n petit (2 ou 3) le développement en série est plus rapide.

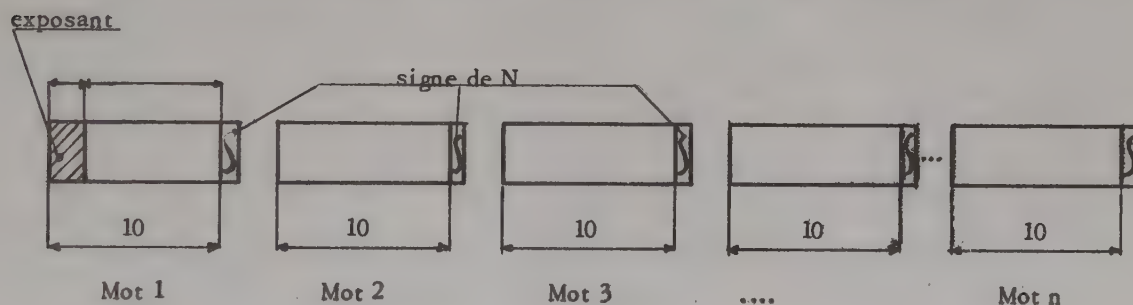
SECTION 1.2

REALISATION D'UNE ARITHMETIQUE POUR NOMBRES REELS

VIRGULE FLOTTANTE MULTIPLE PRECISION

L'Arithmétique a pour but de traiter les nombres VFMP, c'est-à-dire d'effectuer sur ces nombres les opérations arithmétiques suivantes : addition, soustraction, multiplication et division et quelques transferts que nous expliciterons plus loin. La structure de l'Arithmétique est donc étroitement conditionnée par celle des nombres traités qui sont essentiellement formés d'un exposant de p chiffres ($p = 2$ ou 3) et d'une mantisse formée de n tranches de 10 chiffres (sauf la première tranche qui peut ne contenir que les $(10-p)$ premiers chiffres).

En mémoire générale l'exposant est incorporé à la première tranche de la mantisse qui est alors normalisée. Le nombre réel multiple précision d'ordre n , aura donc la forme suivante :



C'est sous cette forme, à " n Mots", qu'un nombre sera chargé sur le tambour ou bien perforé sur carte.

Le Mot 1 est toujours celui qui contient l'exposant ; chaque tranche de 10 chiffres constitue un Mot 650 et occupe la totalité d'une mémoire adressable.

1.21 • DISPOSITION DES NOMBRES EN MEMOIRE

De la structure à " n Mots", il résulte que pour traiter chaque nombre il faudra faire connaître au programme d'Arithmétique l'adresse des n mémoires contenant les n Mots du nombre ; toute opération portant sur 2 nombres exigera donc la connaissance de $2n$ adresses. A cette disposition, nous en avons préféré une autre plus systématique.

Supposons que nous ayons k nombres à n Mots à traiter.

En premier lieu nous chargerons les k Mots 1 des k nombres dans k mémoires définies d'adresses $F_1, F_2 \dots F_k$.

Puis, nous chargerons tous les k Mots 2 dans des mémoires toutes situées à une même distance D' de la mémoire F_1 contenant le Mot 1 du même nombre. Ces Mots 2 seront donc chargés dans des mémoires d'adresses :

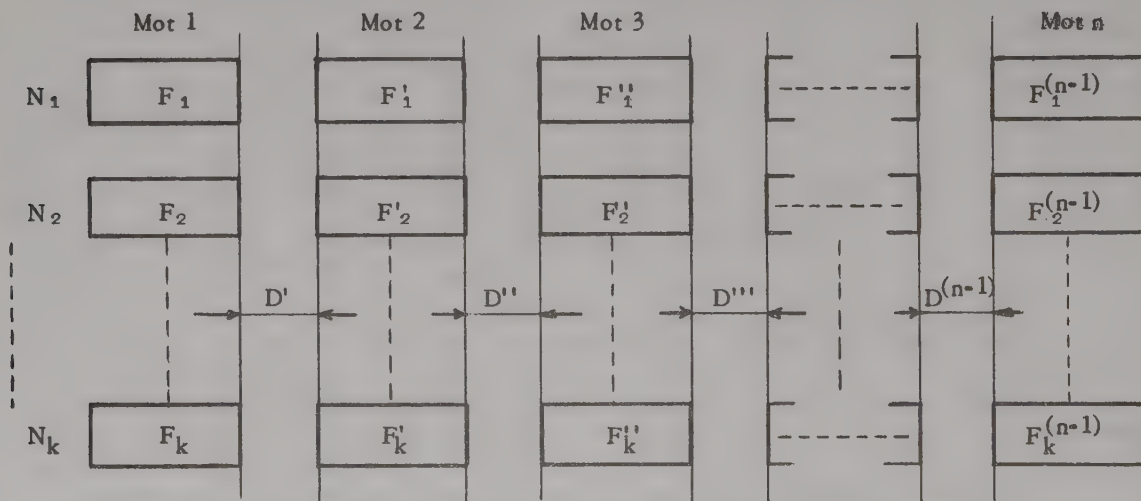
$$F'_1 = F_1 + D', F'_2 = F_2 + D' \dots F'_k = F_k + D'$$

On appelle "Distance" ou encore "Constante de translation" entre 2 mémoires d'adresses F et F' ($F' > F$) la différence positive $F' - F$.

De même tous les Mots 3 seront tous chargés dans des mémoires toutes situées à une même distance D'' des mémoires F' contenant le Mot 2 du même nombre.

$$\begin{aligned} F''_1 &= F'_1 + D'' = (F_1 + D') + D'' \\ &\dots\dots\dots \\ F''_k &= F'_k + D'' = (F_k + D') + D'' \end{aligned}$$

et ainsi de suite pour les Mots 4, 5 ... n toujours tous situés à une même distance du Mot précédent. La disposition des k nombres chargés en mémoire sera donc schématiquement la suivante :



En utilisant une telle méthode il ne suffit plus que de fournir les k adresses $F_1, F_2 \dots F_k$ des Mots 1 et les $(n - 1)$ distances entre les Mots consécutifs d'un même nombre. A partir de ces $(k + n - 1)$ données, le Programme d'Arithmétique se chargera lui-même de construire les adresses $F'_1, F'_2 \dots F'_k$ $F^{(n-1)}_1, F^{(n-1)}_2 \dots F^{(n-1)}_k$ des Mots 2 Mots n de ces k nombres.

Ce procédé a l'avantage de systématiser la disposition des nombres en mémoire et de simplifier la programmation puisqu'une partie de cette dernière se trouve prise en charge par l'Arithmétique elle-même.

Remarque : Les quantités $D', D'' \dots D^{(n-1)}$ peuvent être ou non égales entre elles. Elles sont positives. En aucun cas elles ne peuvent être nulles jusqu'à $D^{(n-1)}$ inclus.

1.22 - PSEUDO - BLOC DE CALCUL

Avant de pouvoir définir la structure générale d'une Arithmétique, il est nécessaire de déterminer le schéma d'une opération dans cette Arithmétique.

1.221 - SCHEMA D'UNE OPERATION SUR 650

Toute opération arithmétique se fait dans l'Accumulateur 8002 - 8003 par l'intermédiaire du distributeur 8001, de l'additionneur (non adressable) et exige la programmation de 2 instructions.

Soit par exemple l'opération : $A + B = C$

A et B étant 2 opérandes chargés dans les mémoires d'adresses M_A, M_B du tambour et n'occupant chacun qu'un seul Mot.

L'accumulateur sert à enregistrer l'un des opérandes avant opération, A par exemple, et reçoit le résultat C après opération. Le distributeur est un intermédiaire pour les opérations arithmétiques, toute opération déclenchée par le 650 commençant par transférer dans le distributeur celui des 2 opérandes qui est encore en mémoire générale, soit B, l'opération s'effectuant ensuite entre l'accumulateur et le distributeur par l'intermédiaire de l'additionneur. Il y a donc 4 phases :

- 1ère phase : charger A dans 8002 - 8003
- 2ème phase : charger B dans 8001
- 3ème phase : effectuer l'opération par l'intermédiaire de l'additionneur.
- 4ème phase : le résultat C apparaît en 8002 - 8003,

les 3 dernières phases ne nécessitant qu'une seule instruction.

1.222 - SCHEMA D'UNE OPERATION DANS L'ARITHMETIQUE

Pour effectuer une opération en Arithmétique, VFMP on conservera le même schéma en 4 phases en l'adaptant simplement à la structure des nombres à traiter.

L'accumulateur 8002 - 8003 et le distributeur ne sont plus de capacité suffisante et on leur substitue 2 groupes de mémoires générales ayant un rôle absolument identique et que l'on appelle "Pseudo-Accumulateur" et "Pseudo-Distributeur", en abrégé PA et PD. Le rôle de l'additionneur sera tenu par les différents sous-programmes d'opérations.

Les 4 phases d'une opération seront donc :

- 1ère phase : charger A dans le "Pseudo-Accumulateur"
- 2ème phase : charger B dans le "Pseudo-Distributeur"
- 3ème phase : effectuer l'opération par l'intermédiaire du sous-programme approprié.
- 4ème phase : le résultat C apparaît sous forme normalisée dans le "Pseudo-Accumulateur".

1.223 - LE PSEUDO-ACCUMULATEUR

Le Pseudo-Accumulateur d'une Arithmétique VFMP est constitué de $(n + 1)$ mémoires générales adressables appelées $K_0, K_1, K_2, \dots, K_i, \dots, K_n$ et contenant l'opérande $A = (\alpha, a)$ sous la forme $(n+1)$ Mots :

(K_0)	$= 00000 \dots \alpha \dots$	exposant de A
(K_1)	$= 000 \dots \leftarrow a_1 \rightarrow$	Mot 1
(K_2)	$= \leftarrow a_2 \rightarrow$	Mot 2
:	:	
(K_i)	$= \leftarrow a_i \rightarrow$	Mot i
:	:	
(K_n)	$= \leftarrow a_n \rightarrow$	Mot n

Le Pseudo-Accumulateur contiendra sous la forme précédente, l'un des opérandes avant opération et le résultat après opération.

1.224 - LE PSEUDO - DISTRIBUTEUR

Le Pseudo-Distributeur d'une Arithmétique VFMP est constitué de $(n + 1)$ mémoires générales adressables, appelées $H_0, H_1, H_2, \dots H_i \dots H_n$ et contenant l'opérande $B = (\beta, b)$ sous la forme $(n + 1)$ Mots :

(H_0)	$= 0000 \dots \beta \dots$	exposant de B
(H_1)	$= 000 \leftarrow b_1 \rightarrow$	Mot 1
(H_2)	$= \leftarrow b_2 \rightarrow$	Mot 2
:	:	
(H_j)	$= \leftarrow b_j \rightarrow$	Mot j
:	:	
(H_n)	$= \leftarrow b_n \rightarrow$	Mot n

Le Pseudo-Distributeur contiendra sous la forme précédente et au cours de l'opération, le second opérande, qui avant opération est chargé en mémoire générale. Après opération et sauf dans le cas de la division, il contient encore le second opérande sous cette forme.

1.225 - SOUS - PROGRAMMES D'OPERATIONS

Ces sous-programmes sont au nombre de 3 :

- 1 - Sp d'addition dans lequel sont groupées les 4 opérations : $A + B$; $A - B$ et $A + |B|$ et $A - |B|$
- 2 - Sp de multiplication qui effectue $A \times B$
- 3 - Sp de division qui effectue $A : B$ et fait appel aux deux premiers dans le cas général.

1.23 • SCHEMA DE L'ARITHMETIQUE

1.231 - PRESENTATION DES DONNÉES A L'ENTREE

Les nombres à traiter sont chargés en mémoire - mémoire générale ou intermédiaire d'entrée - sous la forme " n Mots ". Dans toute opération, l'un des opérandes doit être initialement chargé dans le P.A.; en début de calcul ou à chaque nouveau calcul, il faut donc prévoir une opération de transfert vers le P.A.

Une instruction de calcul doit exprimer la nature de l'opération à effectuer et indiquer l'adresse F du Mot 1 du nombre à traiter ainsi que l'adresse I de l'instruction à suivre; les adresses des Mots 2 ... n étant calculées par l'arithmétique, Tous ces renseignements sont fournis à l'entrée de l'Arithmétique qui les stocke et les exploite en vue de l'opération à effectuer.

L'Arithmétique est divisée en 4 groupes de sous-programmes :

- 1 • Sous-programmes d'entrée : SPE
- 2 • Sous-programmes de Transfert : SPT
- 3 • Sous-programmes d'opérations : SPO
- 4 • Sous-programmes de Sortie : SPS

1.232 - PRESENTATION DES RESULTATS A LA SORTIE

Au cours des différents sous-programmes, exposant et mantisse sont séparés et traités séparément. A la sortie de l'Arithmétique le résultat apparaîtra sous 2 formes :

- Forme (n + 1) Mots dans le P.A (exposant et mantisse séparés).
- Forme n Mots (exposant et mantisse regroupés).

Mot 1 en 8003

Mot 2 en 8002 et Mot 3 ... Mot n en $K_3 \dots K_n$

Le fait de faire apparaître une partie du résultat en 8002 - 8003 permet de le tester en fin d'opération.

Le Mot 1 du résultat, mis sous la forme " n Mots ", est également chargé dans une mémoire de travail appelée K'_1 :

$$(K'_1) = (., \gamma .. \leftarrow c_1 \rightarrow)$$

et où il sera repris dans le cas d'un transfert du P.A. vers la mémoire générale.

1.233 - SOUS - PROGRAMMES D'ENTREE

Les différentes opérations réalisées sont les suivantes :

- 1 • Charger les constantes de translation sous la forme 00 D⁽¹⁾ 0000 + dans des mémoires de travail (D⁽¹⁾). Cette opération n'a lieu, en principe, qu'au début des calculs ou à chaque fois que l'une quelconque de ces constantes doit être modifiée.
- 2 • Orientation vers les SPT et SPO : A toute opération correspond une entrée spécifique ; les premières instructions consistent à charger la 1ère instruction du SPT ou SPO correspondant à l'opération dans une

mémoire d'orientation où elle sera reprise à la sortie du SPE.

- 3 - Mettre l'instruction de sortie (I) en mémoire de réserve où on la reprendra à la sortie de l'Arithmétique.
- 4 - Former les instructions d'appel du second opérande; pour cela, placer les adresses $F, F', F'' \dots F^{(n-1)}$ en position d'adresse facteur de ces instructions. On utilise les constantes $D^{(i)}$ et l'adresse F fournies à l'entrée.
- 5 - Appel du contenu des adresses $F, F' \dots F^{(n-1)}$: Mot 1 en 8003, Mot 2 en 8002 et les autres en $K_3, K_4 \dots K_n$.
- 6 - Retour à la mémoire d'orientation.

1.234 - SOUS - PROGRAMMES DE TRANSFERT

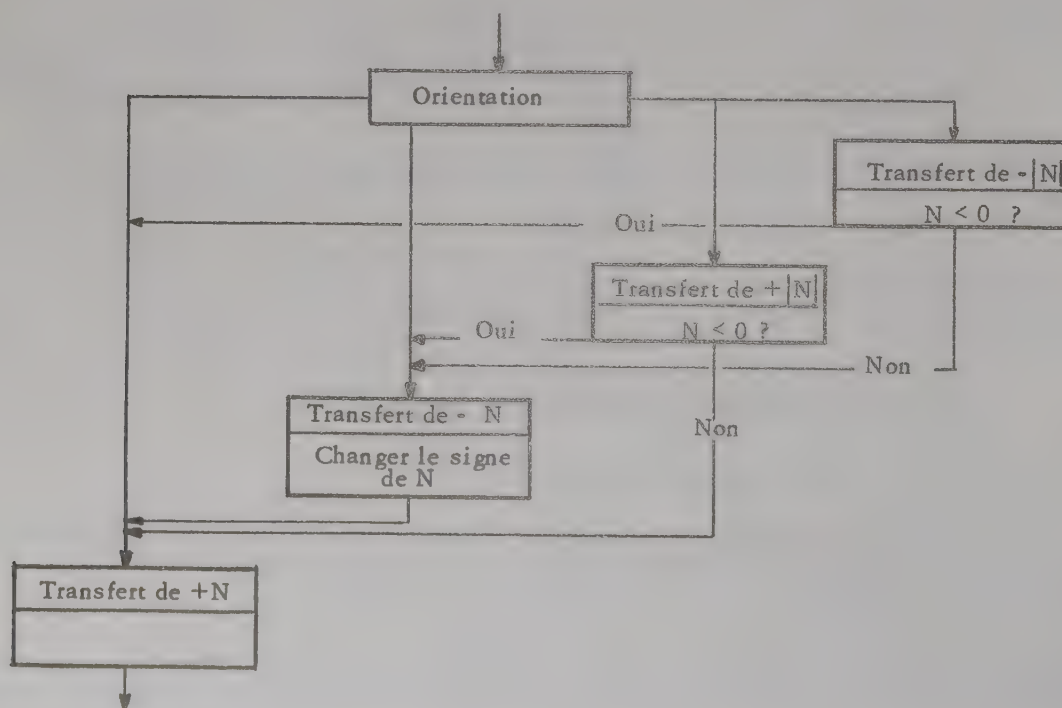
1.234.1 - Transferts vers le P.A.

Le transfert se fait sous la forme $(n+1)$ Mots, exposant en K_0 , la première tranche de la mantisse normalisée avec zéros à la place de l'exposant en K_1 et les autres tranches en $K_2, \dots K_n$. Le transfert peut se faire de sorte que, si N est le nombre chargé en $F, F' \dots F^{(n-1)}$, on puisse transférer dans le P.A., $+N, -N, +|N|$ ou encore $-|N|$.

Mais puisque :

$$\begin{aligned} -N &= +(-N) \\ +|N| &= -N \text{ si } N \text{ est négatif et } +N \text{ si } N \text{ est positif} \\ -|N| &= +N \text{ si } N \text{ est négatif et } -N \text{ si } N \text{ est positif,} \end{aligned}$$

après un changement judicieux du signe de N , on se ramène au transfert de $+N$. 8003 et 8002 contenant les Mots 1 et 2, le sous-programme a le schéma suivant:



Transfert de +N vers le P.A.

• Le Mot 1 est transféré en K'_1 en mémoire de travail en vue de permettre un transfert éventuel du contenu du P.A. vers la mémoire générale (voir ci-dessous).

• Séparation de l'exposant v chargé en valeur absolue en K_0

$$(K_0) = |000 \dots v \ v|$$

• Les différentes tranches de la mantisse vers les K_i

$$(K_1) = 000 \dots n_1 \dots$$

$$(K_2) = \dots n_2 \dots$$

:

$$(K_n) = \dots n_n \dots$$

• Instruction (I) de sortie de l'Arithmétique.

1.234.2 • Transfert vers les mémoires

On transfère sous la forme n Mots et dans les mémoires $F, F' \dots F^{(n-1)}$ le résultat d'une opération déjà contenu sous cette forme dans K'_1 et $K_2, K_3 \dots K_n$.

1 • On construit $F', F'' \dots F^{(n-1)}$ à partir de F et des $D^{(i)}$ que l'on place en position d'adresse facteur des instructions de transfert.

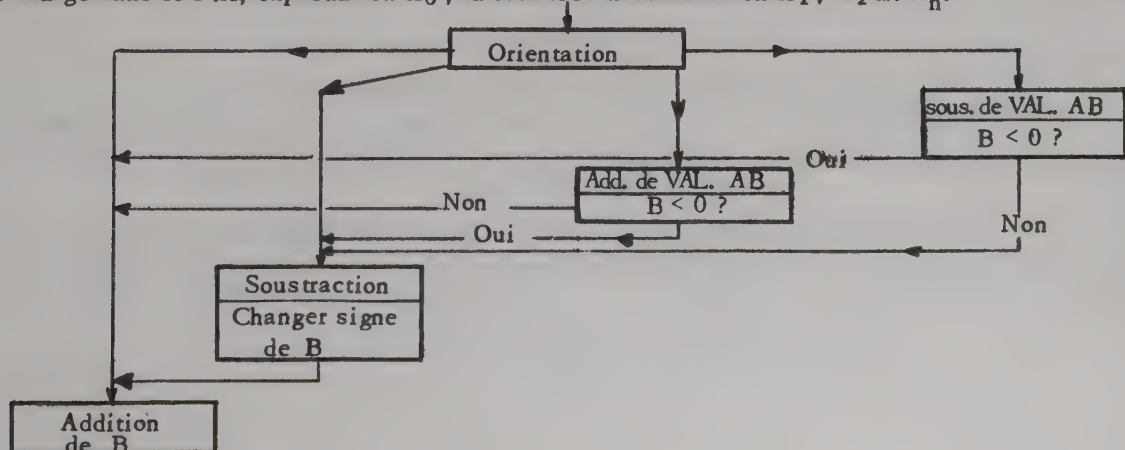
2 • Appels successifs de K'_1 et de $K_2 \dots K_n$ et transfert vers la mémoire correspondante.

3 • Instruction (I) de sortie de l'Arithmétique.

1.235 • SOUS • PROGRAMMES D'OPERATIONS

1.235.1 • Sous-programmes d'addition

Il groupe 4 opérations : l'addition $A + B$ et les 3 opérations qui s'en déduisent $A - B$; $A + |B|$; $A - |B|$ et se ramènent à la première après un changement judicieux du signe de B . A est supposé déjà chargé dans le P.A., exposant en K_0 , la mantisse normalisée en $K_1, K_2 \dots K_n$.



Règles d'addition

Etant donné $A = (\alpha, a)$ et $B = (\beta, b)$ l'addition $A + B = C$ définit le nombre $C = (\gamma, c)$ tel que :

$$\gamma = \text{Sup}(\alpha, \beta)$$

$$c = a \times 10^{-(\gamma - \alpha)} + b \times 10^{-(\gamma - \beta)}$$

Le premier opérande A se trouve déjà dans le P.A, sous la forme $(n+1)$ Mots, la mantisse étant normalisée.

Opérations du sous-programme

- 1 - Charger le second opérande B , après un éventuel changement de signe dans le P.D sous la forme $(n+1)$ Mots ; mantisse normalisée

$$(H_0) = 000 \dots |\beta| \quad (H_1) = 000 \leftarrow b_1 \rightarrow$$

$$(H_2) = \leftarrow b_2 \rightarrow \dots \quad (H_n) = \leftarrow b_n \rightarrow$$

- 2 - Elimination des cas particuliers :

Si $A = 0$, $C = B$; le résultat B doit être transféré dans le P.A. sous la forme $(n+1)$ Mots.

Si $B = 0$, $C = A$; le résultat est déjà chargé dans le P.A.

- 3 - Détermination de γ : on teste la différence $\alpha - \beta$

Si $\alpha - \beta = 0$ $\gamma = \alpha = \beta$ γ est déjà chargé en K_0

Si $\alpha - \beta < 0$ $\gamma = \beta$ il faut charger β en K_0

Si $\alpha - \beta > 0$ $\gamma = \alpha$ γ se trouve déjà en K_0 .

- 4 - Calcul de c . 3 cas sont à considérer :

1er cas : $\gamma = \alpha = \beta$

donc $\gamma - \alpha = \gamma - \beta = 0$ et $c = a + b$

Il n'y a aucun décalage à faire ; on passe à l'addition de 2 mantisses, a chargée dans le P.A et b dans le P.D (Voir annexe A au 1.245.1).

2ème cas : $\gamma = \beta$

donc $\gamma - \beta = 0$ et $\gamma - \alpha = \beta - \alpha$ donc $c = a \times 10^{-(\beta - \alpha)} + b$.

On décale d'abord a de $|\beta - \alpha|$ positions vers la droite (voir annexe B au 1.245.1) , puis a décalé ayant été rechargé dans le P.A , on passe à l'addition (cf. Annexe A au 1.245.1).

3ème cas : $\gamma = \alpha$

donc $\gamma - \alpha = 0$ $\gamma - \beta = \alpha - \beta$ et $c = a + b \cdot 10^{-(\alpha - \beta)}$

On décale d'abord b de $|\alpha - \beta|$ positions vers la droite (voir Annexe B au 1.245.1) puis b décalé ayant été rechargé dans le P.D, on passe à l'addition (voir Annexe A au 1.245.1).

- 5 - Le résultat non normalisé de l'addition étant chargé dans le P.A, la suite des instructions est prise à l'entrée du S.P.S en vue du recadrage de c et des corrections possibles de γ ; 8003-8002 contient les 2 premières tranches c_1 et c_2 .

Annexe A au 1.235.1

Addition des 2 mantisses a et b

Nous supposons que ces 2 mantisses sont chargées dans le P.A. et le P.D. après avoir été décalées si nécessaire.

1 - Processus de l'addition

Chaque mantisse se compose de n tranches de 10 chiffres, sauf la première qui ne comporte que (10 - p) chiffres. Pour éviter les risques de dépassement de capacité en cours d'addition, on est contraint à n'utiliser que 8002 pour y effectuer les additions et à réserver 8003 pour y recevoir un report éventuel.

L'addition se fera donc tranche par tranche dans 8002 et le processus est le suivant :

On commence par calculer dans 8002 la somme partielle de rang n, $t_n = a_n + b_n$.

Le contenu de 8002 (r_n) est alors transféré en K_n . Avant d'effectuer l'addition de a_{n-1} et b_{n-1} , on reporte le contenu de 8003 dans 8002 afin d'y inclure le report éventuel de l'addition précédente. La somme de ces 3 termes, qui constitue la somme partielle de rang $n-1$, est alors transférée en K_{n-1} . Et ainsi de suite on forme :

$$r_i \text{ contenu en 8002} = \text{report de } r_{i+1} + a_i + b_i$$

que l'on transfère en K_i , jusqu'à la somme partielle de rang 1. $r_1 = \text{report de } r_2 + a_1 + b_1$ que l'on transfère en K_1 .

Le résultat final de l'opération serait donc :

$$c = \sum_{i=1}^n r_i \times 10^{10i}$$

2 - Correction des r_i

Comme les r_i ont été calculés indépendamment les uns des autres, ils n'ont pas nécessairement le même signe. Ce cas se présente lorsque, a et b étant de signes contraires et que par exemple $|a| > |b|$ il existe des a_i et b_i tels que $|a_i| < |b_i|$, de sorte que $|a| - |b|$ est positif alors que $|a_i| - |b_i|$ est négatif.

Par exemple, si, dans un système de 5 tranches à 3 chiffres, nous considérons les 2 mantisses:

$$a = 216 \ 333 \ 333 \ 333 \ 616 +$$

$$b = 102 \ 333 \ 333 \ 333 \ 800 -$$

Les sommes partielles calculées seront :

$$r_5 = 616 - 800 = 184 -$$

$$r_4 = 333 - 333 = 000 + = r_3 = r_2$$

$$r_1 = 216 - 104 = 114 +$$

de sorte que l'on aurait :

$$a + b = 114 + 000 + 000 + 000 + 184 -$$

Pour tenir compte de cette différence de signe et aussi du fait que des r_i peuvent être nuls, il sera nécessaire de comparer le signe des r_i à celui du résultat et éventuellement de corriger ces sommes partielles.

Le signe du résultat est celui de r_k ou si celui-ci est nul, ce sera le signe de la première des sommes partielles non nulle dans la suite $r_2, r_3 \dots r_k, \dots r_i, \dots r_n$. Supposons que r_k soit la première somme partielle non nulle et que r_i soit la première somme partielle dont le signe est différent de celui de r_k . Deux cas peuvent alors se présenter:

1er cas : $r_k > 0$ et $r_i < 0$

Deux possibilités peuvent également se présenter suivant que r_{i-1} est nul ou non.

1ère possibilité : $r_{i-1} \neq 0$. On place r_{i-1} dans 8003. L'accumulateur reçoit le signe + puis on additionne r_i dans 8002. r_i retrouve alors sa véritable valeur ainsi que le signe du résultat ; r_{i-1} est corrigé et ces deux sommes partielles sont transférées en K_i et K_{i-1} .

2ème possibilité : $r_{i-1} = r_{i-2} = \dots = r_{i-q+1} = 0 ; r_{i-q} \neq 0$. On place une unité positive de rang $i-1$ dans 8003 avant d'additionner r_i dans 8002 qui apparaît alors en vraie valeur et est transféré en K_i . Mais il faut retrancher cette unité positive de rang $i-1$ et puisque $r_{i-1} = r_{i-2} = \dots = r_{i-q+1} = 0, (i-q > k)$, on transfère 999999999 + dans $K_{i-1}, K_{i-2} \dots K_{i-q+1}$ et on retranche une unité positive de rang $i-q$ de r_{i-q} qui est alors transféré en K_{i-q} . En effet :

	$i-q$	$i-q+1$	$i-q+2$			$i-2$	$i-1$	
		99 ... 9	99 ... 9	9 .	9	99 ... 9	99 ... 9	+
+	00 ... 01	00 ... 0	00 ... 0	0	0	00 ... 0	00 ... 0	-
=	00 ... 0	00 ... 0	00 ... 0	0	0	00 ... 0	00 ... 01	-

2ème cas : $r_k < 0$ et $r_i > 0$

1ère possibilité : $r_{i-1} \neq 0$. On place r_{i-1} dans 8003 (accumulateur négatif) puis on additionne r_i dans 8002. Ces 2 sommes corrigées sont transférées en K_{i-1} et K_i .

2ème possibilité : $r_{i-1} = r_{i-2} = \dots = r_{i-q+1} = 0 ; r_{i-q} \neq 0$. On place une unité négative de rang $i-1$ dans 8003 avant d'additionner r_i dans 8002, qui apparaît alors en vraie valeur et est transféré en K_i . On charge 999999999 - dans $K_{i-1}, K_{i-2}, \dots K_{i-q+1}$ et on retranche une unité négative de rang $i-q$ de r_{i-q} qui est alors transféré en K_{i-q} . En effet :

	$i-q$	$i-q+1$	$i-q+2$			$i-2$	$i-1$	
		99 ... 9	99 ... 9	9 .	9	99 ... 9	99 ... 9	-
+	00 ... 01	00 ... 0	00 ... 0	0 .	0	00 ... 0	00 ... 0	+
=	00 ... 0	00 ... 0	00 ... 0	0 .	0	00 ... 0	00 ... 01	+

Une fois la somme partielle r_i corrigée, on compare le signe de $r_{i+1}, r_{i+2} \dots$ à celui de r_k jusqu'à rencontrer une nouvelle somme partielle de signe contraire à celui de r_k . La correction se fait comme précédemment et ainsi de suite jusqu'à épuisement des $(n-i)$ sommes partielles restantes. Lorsque tous les r_i ont été testés et éventuellement corrigés, on a réellement :

$$C = \sum_{i=1}^n r_i^* \cdot 10^{-10i} \quad r_i^* = r_i \text{ corrigé si nécessaire.}$$

Si nous revenons à l'exemple précédent : r_1 est positif, r_5 est négatif et r_2, r_3, r_4 sont nuls. On se trouve dans le premier cas, seconde possibilité. On corrige r_5 :

$$\begin{array}{r} 001\ 000 + \\ \underline{184 -} \\ 000\ 816 + \rightarrow K_5 \end{array}$$

On transfère $999 +$ en K_2, K_3 et K_4 à la place de r_2, r_3 et r_4 et on modifie r_1 :

$$\begin{array}{r} 000\ 114 + \\ \underline{001 -} \\ 000\ 113 + \rightarrow K_1 \end{array}$$

Le résultat étant en effet : $113\ 999\ 999\ 999\ 816 +$

Annexe B au 1.235.1

Décalage d'une mantisse

La différence $|\beta - \alpha| = 00du$ étant en position d'adresse facteur de 8002, le problème consiste à décaler l'ensemble de la mantisse a (ou b) de "du" positions vers la droite, la mantisse décalée devant être arrondie sur sa dernière position. (d représente le chiffre des dizaines et u celui des unités de la différence $|\beta - \alpha|$)

1 - Fractionnement des décalages

Dans une instruction de décalage à droite, de la forme $31\ 00du\ xxxx$ ou $30\ 00du\ xxxx$, le nombre de décalages effectués sur le contenu de 8002 - 8003 est indiqué uniquement par le chiffre "u" en position unité d'adresse facteur ; une seule instruction ne pourra effectuer au maximum qu'un décalage de 9 positions (10 positions pour le code 31, avec $u = 0$). Or la différence $|\alpha - \beta| = 00du$ peut varier de 1 à $10n - p$ et même jusqu'à $10^p - 2$. Il faudra donc fractionner les "du" décalages en tranches de 10 au maximum, soit théoriquement $(d+1)$ tranches.

Mais si les nombres A et B ont des exposants qui diffèrent de plus de $10n - p$ unités, après décalage l'une des mantisses sera complètement décalée par rapport à l'autre, de sorte que dans l'addition des 2 mantisses, cette première ne joue plus aucun rôle ; il n'y a donc effectivement addition que pour autant que $|\alpha - \beta| \leq 10n - p$.

Dans le cas contraire, et suivant le signe de $(\alpha - \beta)$ l'un des 2 nombres sera négligé devant l'autre.

2 - Etude du cas où $\gamma = \beta > \alpha$

1er cas : $\alpha < \beta \leq \alpha + 10$

2ème cas : $\alpha + 10 < \beta \leq \alpha + 20$

.....

$(n-1)$ ème cas : $\alpha + (n-1)10 < \beta \leq \alpha + 10n - p$

n ème cas : $\alpha + 10n - p < \beta$

Pour distinguer ces différents cas, puisque $\alpha - \beta$ est initialement négatif on lui ajoute 10, puis 10, ... puis p et à chaque fois on teste le signe de la nouvelle différence. Remarquons qu'un seul test suffit pour séparer le cas négatif du cas positif ou nul, car en addition, le nombre zéro s'écrit 000 ... 000 + dans 8002 - 8003. Ces 2 derniers cas seront donc traités identiquement puisque l'instruction 31 00do xxxx signifie 10 décalages à droite avec arrondi.

1er cas : $\alpha < \beta \leq \alpha + 10$ $|\alpha - \beta| = u$

Une seule tranche de u décalages avec arrondi sur le Mot n et sans arrondi sur les autres mots. La mantisse ainsi décalée est chargée dans $K_n, K_{n-1} \dots K_1$ en commençant par la droite. Si $|\alpha - \beta| = 10$ donc $u = 0$, le Mot n disparaît, le Mot $n - 1$, éventuellement arrondi, est chargé en K_n , le Mot $n - 2$ en K_{n-1} ... et K_1 sera rempli de zéros.

Cas général : $\alpha + (k - 1) 10 < \beta \leq \alpha + k \cdot 10$

On néglige les Mots $a_n, a_{n-1} \dots a_{n-k+1}$. La tranche a_{n-k} sera décalée de u positions à droite avec arrondi, les tranches $a_{n-k-1}, \dots, a_1$ sans arrondi. La mantisse décalée est alors rechargée dans le PA en commençant par K_n jusqu'à K_{n-k} . Par contre $K_{n-k-1}, \dots, K_1$ sont remplis de zéros. Si $|\alpha - \beta| = 10k$ le décalage de 10 positions avec arrondi se fait sur a_{n-k} et a_{n-k-1} et les tranches $a_{n-k-2}, \dots, a_1$ sont directement transférées dans $K_{n-1}, K_n \dots K_{n-k}$.

Avant dernier cas : $\alpha + (n-1) 10 < \beta \leq \alpha + 10n - p$

On ne conserve que a_1 que l'on décale de u positions avec arrondi à droite et que l'on charge en $K_n, K_{n-1}, K_{n-2} \dots K_1$ sont chargés de zéros.

Dernier cas : $\alpha + 10n - p \leq \beta$

Le nombre A est négligé devant B et l'on se ramène au cas particulier où $A = 0$.

Une fois a décalé et rechargé dans le PA, on prend la suite des instructions au programme d'addition : cf Annexe A au 1.245.1.

3 - Etude du cas où $\gamma = \alpha > \beta$

Les cas à distinguer sont les suivants :

1er cas : $\beta < \alpha \leq \beta + 10$

2ème cas : $\beta + 10 < \alpha \leq \beta + 20$

.....

$(n - 1)^e$ cas : $\beta + (n - 1) \cdot 10 < \alpha \leq \beta + 10n - p$

n^e cas : $\beta + 10n - p < \alpha$.

On rend $\beta - \alpha$ négatif puis on ajoute 10, puis 10, puis p et à chaque fois on teste le signe de la nouvelle différence. Les décalages se font suivant le même principe sur la mantisse b que l'on recharge ensuite dans le P.D. Considérons le cas général :

$\beta + (k - 1) 10 < \alpha \leq \beta + k \cdot 10$

On néglige $b_n, b_{n-1} \dots b_{n-k+1}$. La tranche b_{n-k} est décalée de u positions avec arrondi ; les tranches $b_{n-k-1}, \dots, b_1$ sans arrondi, puis la mantisse décalée est rechargée en commençant par $H_n, H_{n-1} \dots H_{n-k}$; $H_{n-k-1}, \dots, H_1$ sont chargés de zéros. Si $\alpha = \beta + 10k$, le décalage de 10 positions avec arrondi se fait sur les tranches b_{n-k} et b_{n-k-1} et les tranches $b_{n-k-2},$

..., b_1 sont directement chargées dans $H_{n-1}, \dots, H_{n-k}$.

Dans le dernier cas où $\beta + 10 \cdot p \leq \alpha$, le nombre B est négligé devant A et on se ramène au cas particulier $B = 0$.

Une fois b décalé et rechargé dans le P.D., on prend la suite des instructions au programme d'addition : cf. Annexe A au 1.245.1.

1.235.2 - Sous-programme de Multiplication

Etant donné $A = (\alpha, a)$ et $B = (\beta, b)$, la multiplication $A \times B = C$ définit le nombre $C = (\gamma, c)$ tel que $\gamma = \alpha + \beta - E_0$

$$c = \sum_{l=1}^n \left[\sum_{\substack{i,j=1 \\ i+j=l+1}}^n [G(a_i \times b_j) + Q(a_i \times b_j)] \times 10^{-10l} \right]$$

Le premier opérande A se trouve déjà dans le P.A. sous la forme $(n+1)$ Mots, la mantisse étant normalisée.

1- Initialement contenu dans les mémoires d'adresses $F, F' \dots F^{(n-1)}$ sous la forme n Mots, l'opérande B est chargé dans le P.D. sous la forme $(n+1)$ Mots avec zéros à droite.

$$\begin{aligned} (H_0) &= 000 \dots | \beta | & (H_1) &= \text{---} b_1 \text{---} \\ (H_2) &= \text{---} b_2 \text{---} \dots & (H_n) &= \text{---} b_n \text{---} \dots 000 \end{aligned}$$

2 - Modification de l'opérande A chargé dans le P.A. Les zéros situés à gauche de la mantisse doivent passer à droite de sorte que l'on ait :

$$\begin{aligned} (K_0) &= 000 \dots | \alpha | & (K_1) &= \text{---} a_1 \text{---} \\ (K_2) &= \text{---} a_2 \text{---} \dots & (K_n) &= \text{---} a_n \text{---} \dots 000 \end{aligned}$$

3 - Elimination des cas particuliers : Si $A = 0$, $A \times B = 0$ le résultat est déjà chargé dans le P.A.

Si $B = 0$, le résultat seul doit être chargé dans le P.A qui contient A supposé non nul.

4 - Calcul de γ : γ est défini par la relation $\gamma = \alpha + \beta - E_0$. En fait, on calcule $\gamma' = \alpha + \beta - E_0 - p$ Les mantisses a et b étant cadrées tout à gauche, le produit c le sera également et puisque dans le sous-programme de sortie la position repère est prise p positions à droite de l'extrême gauche, lors du recadrage de c_n , le γ' sera majoré de p unités correspondant aux p positions dont c aura été décalé à droite pour être aligné sur la position repère. On aura bien :

$$\gamma' + p = \gamma$$

5 - Calcul de la mantisse c - c est défini par la relation.

$$c = c_1 \cdot 10^{-10} + c_2 \cdot 10^{-20} + \dots + c_\ell \cdot 10^{-10\ell} + \dots + c_n \cdot 10^{-10n} + [c_{n+1} \cdot 10^{-10(n+1)}]$$

$$\text{avec } c_\ell = \sum_{i,j=1}^n [G(a_i \times b_j) + D(a_i \times b_j)]$$

Le développement de c se faisant jusqu'au terme c_{n+1} dont on ne retiendra qu'un éventuel report sur c_n

- On calcule tous les produits partiels $a_i \times b_j$ tel que $(i+j) = (n+1)$.
- Après chaque calcul, on sépare le produit en $G(a_i \times b_j)$ et $D(a_i \times b_j)$ que l'on transfère dans 2 mémoires de travail différentes. Remarquons que le produit $a \times b$ et tous les produits partiels $a_i \times b_j$ ont même signe.
- Dans 8002 on calcule c_{n+1} .

$$c_{n+1} = \sum_{i,j=1}^{n+1} [G(a_i \times b_j) + D(a_i \times b_j)]$$

dont on ne conserve que le report en 8003 après un décalage avec arrondi de 10 positions à droite. Les termes $c_{n+2} \dots c_{2n}$ sont négligés.

- Dans 8002 on calcule c_ℓ , puis le contenu de 8002 étant transféré dans K_ℓ , avant de faire la somme des termes de rang $(\ell - 1)$, on transfère le report précédent de 8003 dans 8002 et ainsi de suite jusqu'à c_2 et c_1 .

6 - En fin de calcul, c_1 se trouve en 8003, c_2 en 8002 et tous les autres c_ℓ dans les K_ℓ . La suite des instructions est prise à l'entrée du SPS pour la normalisation de c.

1.23 §.3 - Sous-programme de Division

Etant donné $A = (\alpha, a)$ et $B = (\beta, b)$, la division $A : B = C$ définit un nombre $C = (\gamma, c)$ tel que $\gamma \approx \alpha - \beta + E_0$
C est défini par une expansion de division (Voir § 1:123)

Le premier opérande A se trouve déjà dans le P.A. sous la forme $(n+1)$ Mots, la mantisse étant normalisée.

1 - Charger le second opérande B initialement contenu dans les mémoires d'adresses F, F', ... F⁽ⁿ⁻¹⁾ sous la forme $(n+1)$ Mots, zéros à droite dans le P.D.

Modifier la mantisse de A de manière à placer les zéros à droite. Recharger dans le P.A. Ces 2 opérations sont communes avec le sous-programme de Multiplication.

2 • Elimination des cas particuliers. Si $A = 0$. Le résultat est nul et se trouve déjà chargé dans le P.A. Ce cas est considéré en commun avec le sous-programme de Multiplication.

Si $B = 0$. Il y a un arrêt programmé. (cas particulier au sous-programme de Division).

3 • Calcul de γ . γ est défini par la relation $\gamma = \alpha - \beta + E_0$.

En fait, on calcule $\gamma' = \alpha - \beta + (E_0 - p)$. Les mantisses a et b n'étant pas normalisées, la mantisse du quotient ne le sera pas non plus et puisque la position repère est prise p unités à droite de l'extrême gauche, la libération de ces positions dans le S.P.S. se traduira par une majoration du γ' précédemment calculé de p unités de sorte que l'on retrouvera bien γ .

4 • Calcul de c par programmation de l'algorithme manuel.

1 • Au départ de la division. Remarquons que pour simplifier la programmation le calcul de c se fait en valeur absolue, son signe étant déterminé séparément.

La mantisse a constitue le premier dividende partiel D_1 ; les tranches de rang $n+1$, $n+2$... $2n$ de a qui seront prises en considération dans le développement du quotient seront formées de zéros.

D_1 est chargé dans n mémoires désignées par $MA_1, MA_2 \dots MA_n$ et le diviseur b dans les n mémoires $MB_1, MB_2, \dots MB_n$.

1ère phase: Prévission. On charge a_1 et a_2 dans 8003 - 8002 (a_1 et $a_2 \neq 0$) et on effectue leur division par b_1 . Les mantisses a et b ne sont pas normalisées, a étant cadré le premier chiffre sur la position 2 du Mot 1 et b sur la position 1 de manière à éviter un dépassement de capacité.

Soit c_1 la prévission de quotient obtenue. c_1 est chargé dans une mémoire MC_1 .

2ème phase: Vérification. On forme le produit de b par $(-c_1)$ en utilisant un sous-programme de multiplication particulier. Le résultat est une mantisse de $(n+1)$ tranches que l'on chargera dans les n mémoires du P.D. : $H_1, H_2 \dots H_n$ auquel on adjoint une mémoire supplémentaire H_{n+1} pour contenir la $(n+1)$ ème tranche du produit.

Après adaptation, on utilise le sous-programme d'Addition pour calculer le reste

$$R_1 = D_1^* + [b \times (-c_1)]$$

D_1^* étant formé du dividende partiel D_1 augmenté à droite de la tranche supplémentaire de la mantisse a de rang $(n+1)$ et donc formée de zéros.

Le reste R_1 , formé de $(n+1)$ tranches apparaît dans n mémoires du P.A. : $K_1, K_2 \dots K_n$ et la mémoire supplémentaire K_{n+1} .

On teste le signe du reste, c'est-à-dire dans le résultat de l'addition le signe de la première somme non nulle. Si ce signe est négatif, la prévission sur c_1 est trop forte et doit être corrigée. Si ce signe est positif ou si R_1 est nul, la prévission sur c_1 est exacte ; il faut donc calculer exactement R_1 , c'est-à-dire vérifier et éventuellement corriger le signe des sommes partielles dont il est constitué. On poursuit alors la division.

3ème phase: Correction. Nous avons déjà vu (§ 1:123.21) que la correction de c_1 se fait uniquement en le diminuant. On diminue donc c_1 de une unité à droite et c'est avec $c'_1 = c_1 - 1$ chargé en MC_1 que l'on reprend la deuxième phase. On effectuera à nouveau cette vérification jusqu'à ce que le nouveau reste, de négatif devienne positif. La valeur $c_1^{(n \dots)}$ alors chargée en MC_1 est la valeur cherchée pour la tranche considérée du quotient.

2. Poursuite de la division. Après avoir défini le nouveau dividende partiel D_2 , on reprend la suite des 3 phases : prévision, vérification et correction, la tranche suivante du quotient est chargée en MC_2 .

Le nouveau dividende partiel D_2 est formé par le reste R_1 de la division précédente. Ce reste comprend $n + 1$ tranches dont la première est nulle.

Cette tranche est toujours nulle du fait du cadrage initialement prévu pour les mantisses a et b .

Si seule la tranche de rang du reste R_1 est nulle, les tranches 2, 3 ... $n + 1$ sont transférées dans $MD_1, MD_2, \dots, MD_n$ et MD_{n+1} reçoit des zéros.

Si les tranches 1 et 2 sont nulles on effectue les mêmes transferts que précédemment car la tranche correspondante du quotient peut ne pas être nulle en position unité du fait du processus de division dans le 650.

Si les tranches 1, 2 ... j sont nulles ($2 < j < n + 1$), on transfère des zéros dans $MC_2, MC_3, \dots, MC_j$ puisque $c_2, c_3 \dots c_j$ sont nuls et on transfère les tranches non nulles dans les mémoires MD_i : les tranches de rang $j + 1, j + 2 \dots n + 1$ dans $MD_1, MD_2, \dots, MD_{n-j}$ et des zéros dans $MD_{n-j+1} \dots MD_{n+1}$.

Si toutes les tranches sont nulles, la division est terminée et les tranches du quotient à droite de la tranche actuelle sont chargées de zéros.

Ayant donc formé et chargé le nouveau dividende partiel, on reprend la suite des 3 phases de tranche en tranche de quotient jusqu'à la fin de la division.

3. Fin de la division. La division est théoriquement terminée dès qu'un reste R_i devient nul ; dans ce cas $c_{i+1}, \dots, c_n$ ($i < n$) sont nuls. Pratiquement la division sera terminée dès que les n tranches du quotient auront été calculées.

Comme la recherche des différentes tranches du quotient est faite sur la valeur absolue du dividende et du diviseur, il reste à déterminer le signe du reste et celui du quotient : le signe du reste est celui du dividende et celui du quotient est plus, si dividende et diviseur sont de mêmes signes, moins dans le cas contraire.

On charge le quotient affecté de son signe dans le P.A. après avoir transféré le reste avec son signe dans les mémoires MD_i .

5. Calcul de c par développement en série.

c est défini par le produit :

$$c = M \times L \quad \text{avec} \quad M = \frac{a}{b_1} \quad \text{et} \quad L = \sum_{m=0}^n (-f)^m \quad \text{et} \quad f = \frac{\sum_{j=2}^n b_j \cdot 10^{-10j}}{b_1 \cdot 10^{-10}}$$

1. Calcul de M . Le dividende est à n tranches alors que le diviseur est à une tranche. M se calcule donc par expansion du dividende, les quotients partiels étant successivement transférés sur mémoires de travail.

Remarque sur le cadrage de a : Etant donné le processus de division sur le 650 qui débute par

un décalage de une position vers la gauche du dividende, il faut que $|a_1| \sim |b_1|$ soit négatif. Dans le cas contraire il faut initialement décaler a de 1 position à droite et augmenter γ de une unité pour corriger ce décalage.

2 - Calcul de $-f$. On élimine d'abord le cas où $f=0$ ($b_2=b_3=\dots b_n=0$). Et on calcule f par expansion du dividende, b_1 et $b_2, b_3 \dots b_n$ étant chargés à leurs rangs respectifs, donc au départ de l'expansion du dividende. 8003 = 00 0000 0000

$$\begin{array}{l} 8002 = \leftarrow b_2 \rightarrow \\ \text{avec } 8001 = \leftarrow b_1 \rightarrow \end{array}$$

$-f$ est un nombre décimal à n tranches que l'on transfère au fur et à mesure dans des mémoires de travail $\phi_1, \phi_2 \dots \phi_n$ (ϕ_1 peut être chargé de zéros).

3 - Calcul de L . L étant définie par la somme d'une série alternée.

$$L = \sum_{m=0}^n (-f)^m$$

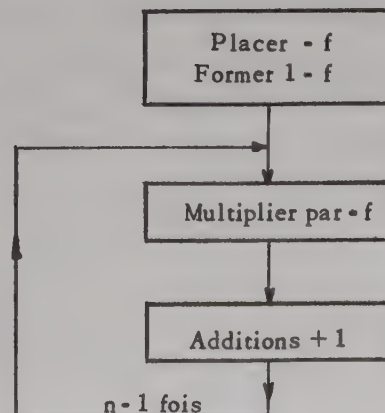
Nous avons vu (§ 1.123.32) qu'il suffit de développer cette somme jusqu'au terme de puissance n , que l'erreur en valeur absolue sur L et par suite sur c faite en négligeant le terme $(-f)^{n+1}$, reste inférieure à $1/2 \times 10^{-(10n-p)}$.

On a donc $L = 1 - f + f^2 - f^3 + \dots + (-f)^n$ pour $n \leq 12$ si $p \leq 3$ et qui s'exprime donc par un polynôme en f .

On calculera donc sa valeur en le développant suivant un schéma de Horner pour lequel tous les coefficients sont égaux à $+1$ pour la valeur $-f$ calculée précédemment. On aura :

	n	$n-1$	$n-2$	...	2	1	0
	1	1	1	...	1	1	1
		$-f$	$-f + f^2$	...		$\sum_{i=1}^{n-1} (-f)^i$	$\sum_{i=1}^n (-f)^i$
$-f$	1	$1 - f$	$1 - f + f^2$	...	$\sum_{i=0}^{n-2} (-f)^i$	$\sum_{i=0}^{n-1} (-f)^i$	$\sum_{i=0}^n (-f)^i$

Ce calcul sera donc programmé sous forme de boucle dont le nombre des itérations est donné par $n-1$



Le sous-programme de Multiplication sera donc utilisé ($n - 1$) fois pour le calcul des différents produits de n tranches par n tranches. Le sous-programme d'addition sera utilisé n fois. L'addition de $+ 1, 000 \dots 000$ se fera en additionnant $0,999 \dots 9 +$ aux n tranches puis $+ 1$ unité de rang n à l'extrême droite. Remarquons qu'aucune difficulté concernant le signe des sommes partielles ne se présentera puisque tous les produits calculés restent inférieurs à 1.

4 • Connaissant M et L en utilisant une dernière fois le sous-programme de multiplication, on calculera c avant de prendre la suite des instructions dans le S.P.S.

6 • Comparaison des 2 méthodes

Il faut considérer séparément le cas des Arithmétiques Double et Triple Précision pour $n = 2$ ou 3 , petit et initialement fixé et le cas d'une Arithmétique Multiple Précision pour n non déterminé à priori et grand $n = 8; \dots 12$.

Dans le premier cas, surtout celui de la Double Précision l'utilisation du développement en série est préférable à tout point de vue. Pour la triple Précision, l'une et l'autre méthode semblent à peu près équivalentes dans le cas général, mais étant donné que les corrections sur les tranches des quotients peuvent être assez nombreuses, le développement en série se révèle beaucoup plus rapide.

Dans le cas d'une Arithmétique multiple précision, pour n grand, comparons ces 2 méthodes. L'une et l'autre méthode se prêtent à une programmation par boucles.

La méthode par développement en série utilise les autres sous-programmes de l'Arithmétique avec un nombre minimum de modifications et elle ne comporte pratiquement pas de tests. Par contre le nombre d'opérations effectuées est important, ces opérations sont longues (surtout les multiplications) bien que dans l'utilisation du sous-programme d'addition, il ne soit pas nécessaire de vérifier le signe de sommes partielles. Le reste de la division n'est pas connu.

La programmation de l'algorithme manuel exige un sous-programme de multiplication particulier. Il comporte de nombreux tests de signe. Par contre, le nombre des opérations arithmétiques est moins important et ces opérations sont moins longues, mais du fait des corrections sur la prévision de la tranche du quotient le nombre de boucles parcourues peut augmenter considérablement. Enfin, les signes du quotient et du reste doivent être déterminés indépendamment.

Cependant cette dernière méthode permet de connaître le reste de la division.

Dans les 2 cas la précision désirée peut être obtenue en développant plus ou moins soit la série, soit le nombre de tranches au quotient.

En fin de compte, dans le cas général, la programmation de l'algorithme manuel se révélerait plus rentable, du point de vue de la rapidité, les encombrements étant analogues et la précision la même pour les 2 méthodes.

1.236 • SOUS-PROGRAMME DE SORTIE

1.236.1 • Recadrage de mantisse

1 • But du sous-programme

Le résultat d'une opération est composé d'un exposant de p chiffres, d'une mantisse de n tranches et doit apparaître sous la forme n Mots.

- Mot 1 : $\gamma.. c_1 \dots$ en 8003 et K'_1
- Mot 2 : $\dots c_2 \dots$ en 8002 et K_1
- Mot n : $\dots c_n \dots$ en K_n .

Pour placer l'exposant à gauche, il faut donc libérer les p positions à gauche de 8003, c'est-à-dire en numérotant de gauche à droite les positions de l'ensemble 8003 - 8002 de 20 à 1, il faut libérer les positions 20, 19, ... 20 - p + 1, la mantisse devant être cadrée sur cette position, c'est-à-dire que son premier chiffre significatif se trouve sur la position 20 - p . - Tout décalage nécessaire de la mantisse pour l'amener sur cette position, tant vers la droite que vers la gauche, introduira une correction équivalente de l'exposant (respectivement en moins et en plus).

2 • Structure de la mantisse en multiplication (ou division)

Les mantisses a et b ayant une structure décimale cadrée en position 20, leur produit présente la même structure, la position 19 étant toujours occupée alors que la position 20 peut l'être ou non. Il faudra donc lui faire subir p ou $p - 1$ décalages à droite avec arrondi et γ augmente d'autant d'unités. Lors de ces décalages, si la mantisse n'est formée que de 9 un report d'unité peut se propager de position en position et venir occuper la position 20 - p + 1 alors qu'elle doit se trouver libre. Cette éventualité est prise en considération lors de la mise en place de l'exposant.

3 • Structure de la mantisse en addition (ou soustraction)

Les 2 mantisses étant initialement cadrées en 20 - p , le résultat l'est également ; seule la position 20 - p + 1 peut être occupée en cas de report. Par contre les positions 20 - p à 1, et les Mots 3 à n peuvent être chargés de zéros précédant le premier chiffre significatif de c . Dans ce dernier cas, il faut décaler c vers la gauche et diminuer γ d'autant.

4 • Recadrage de c

Supposons initialement $c_1 \neq 0$.

On effectue un décalage à gauche avec comptage. Soit Z ce nombre de décalages ; on teste la quantité $Z - p$.

- Si $Z - p < 0$, des positions à gauche de 20 - p étaient occupées.

Il faut donc décaler à droite avec arrondi de $|Z - p|$ positions et augmenter γ de $|Z - p|$ unités.

- Si $Z - p = 0$ il n'y a rien à modifier.

- Si $Z - p > 0$, des positions à droite de 20 - p + 1 sont occupées par des zéros.

Il faut donc décaler à gauche de $|Z - p|$ positions et diminuer γ de $|Z - p|$ unités.

Supposons $c_1 = c_2 = \dots = c_{i-1} = 0$ et $c_i \neq 0$.

On diminue γ de 10 ($i - 1$) puis on porte c_i en 8003, c_{i+1} en 8002, c_{i+2} en $K_2 \dots c_n$ en K_{n-i} , $K_{n-i+1}, \dots, K_n$ sont chargés de zéros, et on reprend le décalage à gauche avec comptage comme précédemment.

1.236.2 • Mise en place de l'exposant

1 • Tests sur la position $20 - p + 1$

En raison du report possible, on teste à nouveau la position $20 - p + 1$ après les décalages avec arrondi. Si elle n'est pas libre, on effectue un nouveau décalage à droite et on majore γ d'une unité.

2 • Tests sur γ

γ doit être un nombre positif de p chiffres.

Si $\gamma \leq 0$, on remplace c par des zéros.

Si $\gamma \geq 10^p - 1$, arrêt overflow.

3 • Mise en place de γ

γ doit être inscrit dans les positions $20, 19, \dots, 20 - p + 1$ à gauche du Mot 1 dans 8003, par addition ou soustraction à la place libérée suivant que c est positif ou négatif. 8003 est alors transféré en K'_1 en vue d'un transfert possible du résultat sur mémoire générale.

4 • Sortie de l'Arithmétique

On retrouve l'instruction I pour l'enchaînement des instructions à la sortie de l'Arithmétique. Le résultat de l'opération apparaissant sous la forme $(n + 1)$ Mots dans le P.A et sous la forme n Mots : en $K'_1, K_2 \dots K_n$ avec également $(8003) = \gamma \dots c_1 \dots$ et $(8002) = \dots c_2 \dots$.

SECTION 1.3

ARITHMETIQUE COMPLEXE

1.31 • STRUCTURE DES NOMBRES COMPLEXES

Etant donné que tout nombre complexe peut se représenter sous la forme d'un couple de nombres réels qui sont sa partie réelle et sa partie imaginaire, il sera possible de représenter un nombre complexe en Virgule Flottante Multiple Précision. A la partie réelle et à la partie imaginaire, on associera un couple exposant-mantisse qui les représente

$$Z = (A, B) = [(\alpha, a), (\beta, b)],$$

α et β étant les exposants de p chiffres et a et b les mantisses de $(10n-p)$ chiffres répartis en n tranches de 10 chiffres.

Le traitement de ces nombres complexes en Arithmétique est analogue à celui des nombres réels; on traitera séparément, sous la forme d'un programme de Calcul en VFMP les relations qui définissent partie réelle et partie imaginaire du résultat d'une opération entre nombres complexes. Rappelons ces relations :

$$\begin{aligned} \text{Addition} \quad (A, B) + (A', B') &= [(A + A'), (B + B')] \\ \text{Soustraction} \quad (A, B) - (A', B') &= [(A - A'), (B - B')] \\ \text{Multiplication} \quad (A, B) \times (A', B') &= [(AA' - BB'), (A'B + AB')] \\ \text{Division} \quad \frac{(A, B)}{(A', B')} &= \left[\left(\frac{AA' + BB'}{A'^2 + B'^2} \right), \left(\frac{A'B - AB'}{A'^2 + B'^2} \right) \right] \end{aligned}$$

Ces relations qui définissent partie réelle et imaginaire du résultat sont des relations entre nombres réels VFMP et seront donc programmées en utilisant l'Arithmétique Réelle de même précision; l'Arithmétique Complexe sera essentiellement constituée des instructions nécessaires à l'utilisation de l'Arithmétique Réelle.

1.32 • DISPOSITION EN MEMOIRE

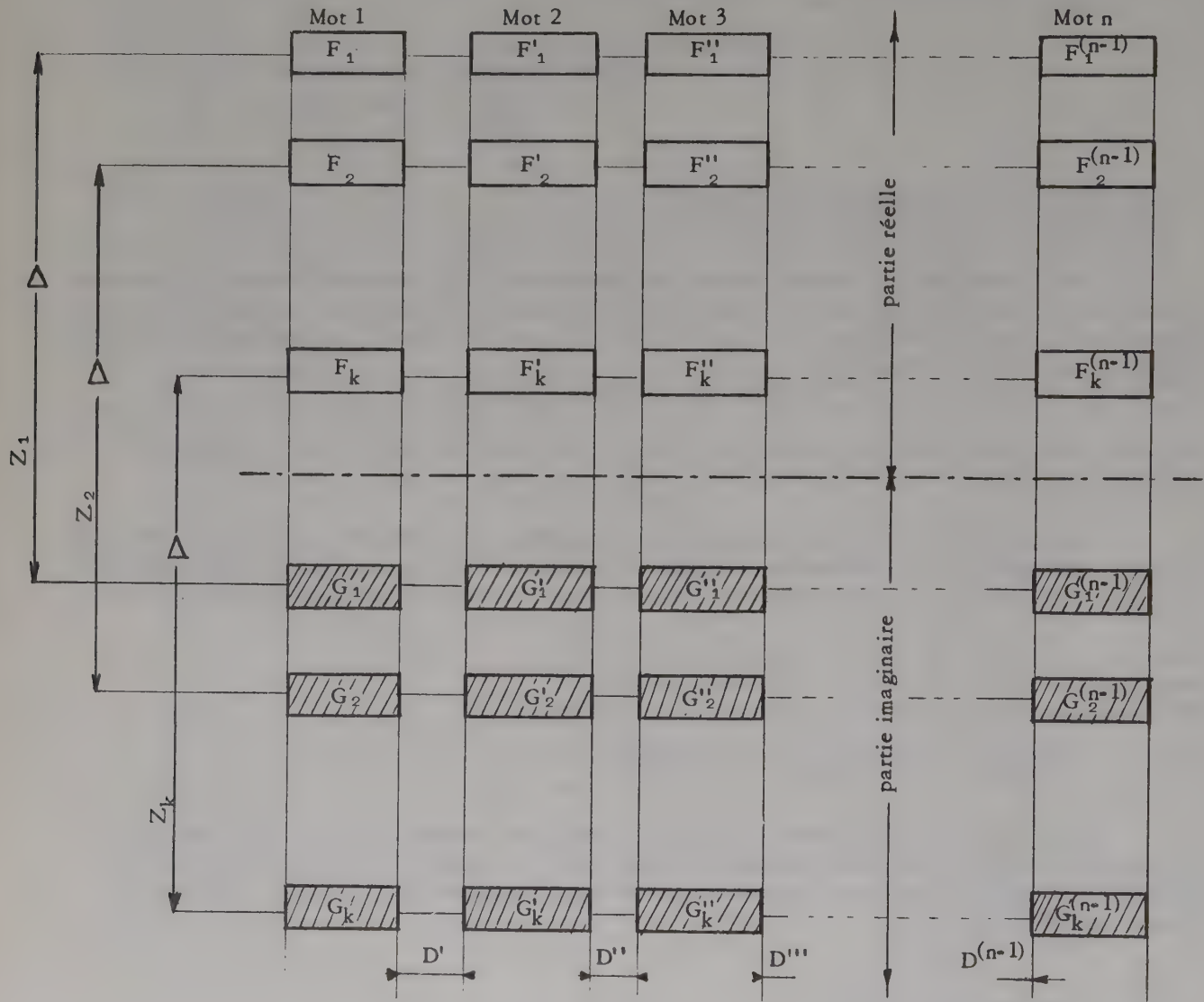
Tout nombre complexe étant représenté par un couple de nombres réels, occupera au moins 2 Mots 650. En simple précision, nous aurons :

$$\begin{aligned} (\text{Mot } 1) &= \text{partie réelle} = A \\ (\text{Mot } 2) &= \text{partie imaginaire} = B \end{aligned}$$

En multiple précision d'ordre n , il faudra 2 séries de n Mots pour représenter un nombre complexe : la 1ère série constituera la partie réelle et la 2ème série, la partie imaginaire du nombre complexe.

En reprenant la disposition systématique, à équidistance de Mots, définie au (§ 1.21.2), nous définirons d'une part les distances $D', D'', \dots D^{(n-1)}$ entre les n Mots de même nature et qui seront les

mêmes dans les 2 séries et d'autre part la distance fixe Δ entre les 2 séries et qui reste la même entre 2 Mots de même rang dans chacune des 2 séries. La disposition en mémoire de k nombres complexes sera donc schématiquement :



La distance entre 2 Mots de la partie réelle d'adresses $F^{(i)}$ et $F^{(i+1)}$ est égale à celle des 2 Mots de la partie imaginaire d'adresses $G^{(i)}$ et $G^{(i+1)}$: $F^{(i+1)} - F^{(i)} = D^{(i)} = G^{(i+1)} - G^{(i)}$ et ceci pour tout nombre Z_j .

Pour des Mots de même rang i , la distance entre partie réelle et partie imaginaire du nombre Z_j est égale à Δ

$$G_j^{(i)} - F_j^{(i)} = \Delta \text{ pour } \forall i, j.$$

Le Mot 1, pris comme repère du nombre Z , est le Mot chargé en F et qui contient l'exposant de la partie réelle de Z . Toutes adresses autres que F sont calculées à partir de cette première adresse et des constantes $D, D' \dots D^{(n-1)}$ et Δ par le sous-programme d'entrée de l'Arithmétique Complexe.

1.33 - PSEUDO - BLOC DE CALCUL

1.331 - SCHEMA D'UNE OPERATION

Les phases d'une opération entre nombres complexes sont les mêmes que dans le cas d'une opération entre nombres réels.

- 1ère phase : Charger le premier opérande complexe dans le Pseudo Accumulateur Complexe : P.A.C._x
- 2ème phase : Charger le second opérande complexe dans le Pseudo Distributeur Complexe : P.D.C._x
- 3ème phase : Effectuer l'opération choisie par l'intermédiaire d'un sous-programme d'opération.
- 4ème phase : Le résultat complexe apparaît, partie réelle et partie imaginaire, sous la forme n Mots dans le P.A.C._x.

On définit donc un Pseudo Accumulateur Complexe (PAC_x) et un Pseudo Distributeur Complexe (PDC_x) formés chacun de 2 séries de n mémoires générales, la 1ère série étant chargée avec la partie réelle et la seconde avec la partie imaginaire du nombre. Remarquons que n mémoires suffisent dans chaque série puisque l'Arithmétique Complexe ne traite les nombres que sous la forme n Mots. La forme $(n+1)$ Mots à exposant et mantisse séparés n'intervient qu'au sein de l'Arithmétique Réelle.

1.332 - PSEUDO - ACCUMULATEUR COMPLEXE

Un nombre complexe $Z = (A, B) = [(\alpha, a), (\beta, b)]$ sera chargé dans le PAC_x sous la forme :

$$\begin{array}{l}
 \text{1ère série} \left\{ \begin{array}{l} (AR_1) = \dots \alpha \dots a_1 \dots \text{Mot}_1 \\ (AR_2) = \leftarrow a_2 \rightarrow \text{Mot}_2 \\ : \\ (AR_n) = \leftarrow a_n \rightarrow \text{Mot}_n \end{array} \right\} \text{ de la partie réelle} \\
 \\
 \text{2ème série} \left\{ \begin{array}{l} (AI_1) = \beta - b_1 \rightarrow \text{Mot}_1 \\ (AI_2) = \leftarrow b_2 \rightarrow \text{Mot}_2 \\ : \\ (AI_n) = \leftarrow b_n \rightarrow \text{Mot}_n \end{array} \right\} \text{ de la partie imaginaire}
 \end{array}$$

Le rôle du PAC_x est de contenir le premier opérande avant opération et le résultat après opération.

1.333 - PSEUDO -DISTRIBUTEUR COMPLEXE

Un nombre complexe $Z' = (A', B') = [(\alpha', a'), (\beta', b')]$ sera chargé dans le PDC_x sous la forme :

$$\begin{array}{l}
 \text{1ère série} \left\{ \begin{array}{l} (DR_1) = \alpha' \dots a'_1 \dots \text{Mot}_1 \\ (DR_2) = \leftarrow a'_2 \rightarrow \text{Mot}_2 \\ : \\ (DR_n) = \leftarrow a'_n \rightarrow \text{Mot}_n \end{array} \right\} \text{ de la partie réelle} \\
 \\
 \text{2ème série} \left\{ \begin{array}{l} (DI_1) = \beta' \dots b'_1 \dots \text{Mot}_1 \\ (DI_2) = \leftarrow b'_2 \rightarrow \text{Mot}_2 \\ : \\ (DI_n) = \leftarrow b'_n \rightarrow \text{Mot}_n \end{array} \right\} \text{ de la partie imaginaire}
 \end{array}$$

Le rôle du PDC_x est de contenir le second opérande au cours de l'opération

1.34 - SCHEMA DE L'ARITHMETIQUE

1.341 - PRESENTATION DES DONNEES A L'ENTREE

Les nombres à traiter sont chargés en mémoire sous la forme n Mots. Dans toute opération l'un des opérands doit être initialement chargé dans le PAC_x . Une instruction de calcul doit exprimer la nature de l'opération à effectuer, et indiquer l'adresse F du Mot 1 du second opérande ainsi que l'adresse I de l'instruction à prendre après l'opération. Les adresses des autres Mots du second opérande seront calculées à partir de F par le sous-programme d'entrée.

Ces renseignements doivent être fournis à l'entrée dans l'Arithmétique Complexe qui les stocke et les exploite en vue de l'opération à effectuer.

L'Arithmétique Complexe ne comporte que 3 groupes de sous-programmes :

- Sous-Programme d'Entrée : SPE
- Sous-Programme de Transfert : SPT
- Sous-Programme d'Opération : SPO

1.342 - PRESENTATION DES RESULTATS A LA SORTIE

A chaque utilisation de l'Arithmétique Réelle, le résultat se présente sous la forme " n Mots" de sorte qu'à la sortie d'une opération Complexe, le résultat apparaîtra nécessairement, partie réelle et partie imaginaire, sous la forme n Mots. Il ne restera plus qu'à le charger sous cette forme dans le PAC_x .

Comme il ne s'agit là que d'un simple transfert, ces opérations ont été directement incluses aux sous-programmes d'opération de sorte qu'à proprement parler, l'Arithmétique Complexe ne comporte pas de sous-programme de Sortie.

1.343 - SOUS - PROGRAMME D'ENTREE

- 1 • Chargement des distances Δ et $D^{(i)}$ sous la forme $00 D^{(1)} 000Q +$ dans les mémoires de travail $DD^{(i)}$. Ces mémoires sont au nombre de n . Cette opération, en général n'a lieu qu'une seule fois en début de programme.
- 2 • Orientation vers les SPT ou SPO : A toute opération sur les nombres complexes correspond une entrée spécifique ; dès la 1ère instruction de cette entrée, la mémoire d'orientation est chargée de la 1ère instruction du SPT ou SPO correspondant à l'entrée.
- 3 • Mise en mémoire de réserve de l'instruction (I) de Sortie de l'Arithmétique Complexe.
- 4 • Formation des instructions d'appel de la partie réelle et de la partie imaginaire du second opérande : on forme les adresses $F, F', \dots F^{(n-1)}$ pour la partie réelle et $G, G' \dots G^{(n-1)}$ pour la partie complexe. F est donné à l'entrée et les distances ont été initialement chargées dans les mémoires $DD^{(i)}$.
- 5 • Appel du contenu de ces adresses et transfert vers le PDC_x : $DR_1, DR_2 \dots DR_n$ et $DI_1 \dots DI_n$.
- 6 • Retour à la mémoire d'orientation.

1.344 - SOUS - PROGRAMME DE TRANSFERT

1.344.1 • Transfert vers le PAC_x

Le transfert se fait partie réelle et partie imaginaire sous la forme n Mots dans les mémoires $AR_1 \dots AR_n$ et $AI_1 \dots AI_n$. On transfère le nombre $Z = (A, B)$ ou bien le nombre $-Z = (-A, -B)$. Ce dernier transfert se faisant comme le premier après avoir changé le signe de la partie réelle et de la partie imaginaire.

Puis on reprend en mémoire de réserve, l'instruction de sortie I .

Remarque : La séparation des exposants des mantisses n'aura lieu qu'au sein de l'Arithmétique réelle lorsque A et B seront utilisés.

1.344.2 • Transfert vers les mémoires générales

Le résultat de l'opération déjà contenu sous la forme n Mots dans le PAC_x est transféré sous cette forme :

- Partie réelle dans les mémoires $F, F', F'' \dots F^{(n-1)}$
 - Partie imaginaire dans les mémoires $G, G' \dots G^{(n-1)}$
- avec $G - F = G' - F' = \dots = G^{(n-1)} - F^{(n-1)} = \Delta$

- 1 • Construire la série des adresses F et G à l'aide de F donné à l'entrée et des distances actuellement chargées dans les mémoires $DD^{(i)}$.
- 2 • Appeler les contenus de $AR_1 \dots AR_n$ que l'on charge dans $F \dots F^{(n-1)}$ puis ceux de $AI_1 \dots AI_n$ que l'on charge dans $G \dots G^{(n-1)}$.
- 3 • Prendre l'instruction de sortie I .

1.345 - SOUS - PROGRAMME D'OPERATION

Dans ces sous-programmes, il s'agit essentiellement de programmer les opérations entre nombres réels qui ont été définis au § 1.31. Puisque l'on utilise une Arithmétique Réelle, il faut donner initialement les distances entre les n Mots de la partie réelle et de la partie imaginaire respectivement chargées dans $AR_1 \dots AR_n$ ou bien $DR_1 \dots DR_n$ et $AI_1 \dots AI_n$ ou bien $DI_1 \dots DI_n$. A chaque utilisation de l'Arithmétique, il faut indiquer l'adresse du Mot 1 du nombre traité : par exemple AR_1 ou AI_1 . Il faut également indiquer l'instruction de sortie de l'Arithmétique Réelle et prendre la suite des instructions à l'entrée de l'Arithmétique Réelle correspondant à l'opération envisagée.

1.345.1 - Addition

L'addition de 2 nombres complexes (A, B) et (A', B') est un nombre complexe dont la partie réelle est $A + A'$ et la partie imaginaire est $B + B'$; (A, B) et (A', B') étant chargés dans le PAC_x et le PDC_x , le sous-programme d'addition est constitué des instructions nécessaires à la préparation de ces opérations entre nombres réels et au transfert du résultat vers le PAC_x .

1 - Calcul de la partie réelle

- + Transfert de A dans le PA de l'Arithmétique réelle - adresse AR_1
- + Addition de A' : adresse DR_1
- + Transfert du résultat de l'opération : adresse AR_1

2 - Calcul de la partie imaginaire

- + Transfert de B dans le PA : adresse AI_1
- + Addition de B' : adresse DI_1
- + Transfert du résultat : adresse AI_1

3 - Le résultat étant transféré en totalité dans le PAC_x , on reprend l'instruction I de sortie de l'Arithmétique Complexe.

1.345.2 - Soustraction

Pour effectuer la soustraction $(A, B) - (A', B')$, on change le signe de A' et B' puis on reprend la suite des instructions comme pour une addition

$$(A, B) - (A', B') = (A, B) + (-A', -B').$$

1.345.3 - Multiplication

Le produit de 2 nombres complexes (A, B) et (A', B') est un nombre complexe dont :

- la partie réelle est : $AA' - BB'$
- la partie imaginaire est : $A'B + AB'$

Le sous-programme de Multiplication est donc essentiellement constitué des instructions nécessaires à la préparation des opérations sur nombres réels définies ci-dessus et résolues au moyen de l'Arithmétique Réelle. (A, B) est chargé dans le PAC_x et (A', B') dans le PDC_x ; le résultat doit l'être dans le PAC_x .

1 • Calcul de la partie réelle

- + Transfert de A dans le PA : adresse AR₁
- + Multiplication par A' : adresse DR₁
- + Transfert provisoire de AA' vers les mémoires de travail T₁
- + Transfert de -B dans le PA : adresse AI₁
- + Multiplication par B' : adresse DI₁
- + Addition de AA' : adresse T₁
- + Transfert provisoire de la somme AA' - BB' vers les mémoires de travail T₁

2 • Calcul de la partie imaginaire

- + Transfert de A' dans le PA : adresse DR₁
- + Multiplication par B : adresse AI₁
- + Transfert provisoire de A'B en mémoire de travail T₂
- + Transfert de A dans le PA : adresse AR₁
- + Multiplication par B' : adresse DI₁
- + Addition de A'B : adresse T₂
- + Transfert de la somme AB' + A'B dans AI₁ ... AI_n puisqu'elle représente la partie imaginaire du résultat.

3 • Rappel de la partie réelle provisoirement transférée en T₁ pour la charger dans AR₁ ... AR_n

4 • Sortie : Prendre l'instruction I.

1.345.4 • Division

Le quotient de 2 nombres complexes (A, B) et (A', B') est un nombre complexe dont :

• la partie réelle est
$$\frac{AA' + BB'}{A'^2 + B'^2}$$

• la partie imaginaire est
$$\frac{A'B - AB'}{A'^2 + B'^2}$$

Remarquons que les numérateurs de la partie réelle et de la partie imaginaire sont les parties réelle et imaginaire du produit (A, B) × A', -B') de sorte que le sous-programme de division fera également appel au sous-programme de multiplication.

Le sous-programme de division est essentiellement formé des instructions nécessaires à la préparation des opérations sur nombres réels définies ci-dessus et à leur résolution au moyen de l'Arithmétique Réelle correspondante. (A, B) est chargé dans le PAC_x, (A', B') dans le PDC_x et le résultat doit apparaître dans le PAC_x.

1 • Calcul du dénominateur A'² + B'²

- + Transfert de A' dans le PA : adresse DR₁
- + Multiplication par A : adresse DR₁
- + Transfert provisoire de A'² dans les mémoires T₃
- + Transfert de B' dans le PA : adresse DI₁

- + Multiplication par B' : adresse DI_1
- + Addition de A'^2 : adresse T_3
- + Transfert provisoire de la somme $A'^2 + B'^2$ en mémoire de travail T_3

2 • Utilisation du sous-programme de Multiplication

- + Transfert de $-B'$ à la place de B' dans $DI_1 \dots DI_n$
- + Modification de la sortie du sous-programme de multiplication pour orienter la suite des opérations sur la fin du sous-programme de division.
- + Calcul des quantités $(AA' + BB')$ et $(A'B - AB')$ au moyen du sous-programme de multiplication et qui sont respectivement chargées dans $AR_1 \dots AR_n$ et $AI_1 \dots AI_n$

3 • Fin de la division

On divise chacune de ces 2 quantités par le dénominateur chargé en T_3 .

Partie réelle

- + Transfert de $AA' + BB'$ dans le PA : adresse AR_1
- + Division par le dénominateur : adresse T_3
- + Transfert du quotient en $AR_1 \dots AR_n$

Partie imaginaire

- + Transfert de $A'B - AB'$ dans le PA : adresse AI_1
- + Division par le dénominateur : adresse T_3
- + Transfert du quotient en $AI_1 \dots AI_n$.

4 • Reprise de l'instruction de sortie I

CHAPITRE II

Arithmétiques réalisées

INTRODUCTION

L'étude de toute méthode de calcul commence par l'écriture d'un premier programme, puis en fonction des résultats obtenus et des qualités requises, cette première ébauche est progressivement améliorée jusqu'à se présenter finalement sous une forme qui soit la meilleure possible.

Dans la réalisation de nos Arithmétiques, nous avons été guidés par trois qualités d'importance croissante :

- rapidité des opérations,
- encombrement minimum du programme,
- précision des résultats.

D'une part il était nécessaire que le résultat d'une opération soit connu avec la précision maximum. Tous les chiffres significatifs devaient être exacts, c'est-à-dire que l'erreur faite sur la mantisse du résultat devait rester inférieure à une demi-unité du dernier ordre.

D'autre part, étant donné le nombre limité des mémoires dont dispose l'IBM 650, il fallait que les programmes réalisés fussent aussi condensés que possible de manière à pouvoir réserver la majeure partie de la mémoire générale aux instructions du programme de calcul faisant appel à l'Arithmétique.

Enfin, à partir du moment où l'on fait appel à une Arithmétique Multiple Précision, la rapidité des opérations passe au second plan. Cependant dans la mesure où les impératifs de précision étaient respectés et où l'encombrement de l'Arithmétique restait raisonnable, il fallait programmer les opérations de manière aussi rapide que possible.

Les Arithmétiques programmées sont :

1) Pour nombres réels

- Arithmétique Double Précision ... (A₀ D, P)
- Arithmétique Mixte Simple • Double précision ... (A .SD .P)
- Arithmétique Triple Précision ... (A .T .P)

2) Pour nombres complexes

- Arithmétique Complexe Simple précision ... (A .Cx .S .P)
- Arithmétique Complexe Double Précision ... (A .Cx .D .P)
- Arithmétique Complexe Mixte Simple • Double Précision ... (A .Cx .SD .P)

Dans ce chapitre, pour chacun de ces 6 programmes, nous avons étudié :

- Les nombres traités par l'Arithmétique,
- Le Pseudo • Bloc de Calcul de l'Arithmétique

et nous avons donné :

- Une notice d'utilisation pour une programmation directe en langage PASO
- Le temps de calcul pour chacune des opérations.

D'autre part, de manière systématique, nous renvoyons l'analyse des instructions de l'Arithmétique au Chapitre 5 : Annexes 1 à 6.

A la fin du chapitre 2, deux exemples de programmation illustrent ces considérations sur l'emploi des Arithmétiques réalisées.

SECTION 2.1

ARITHMETIQUES REELLES

2.11 - ARITHMETIQUE DOUBLE PRECISION (ADP)

L'Arithmétique Double Précision est un programme dont le but est de traiter les nombres Virgule Flottante Flair Double Précision et d'effectuer sur ces nombres l'une des 11 opérations suivantes :

6 opérations arithmétiques : Addition, Soustraction, Addition de valeur absolue, Soustraction de valeur absolue, multiplication et division.

5 transferts : transfert vers la mémoire générale, transfert vers le PA du nombre considéré, de son opposé, de sa valeur absolue ou de l'opposé de sa valeur absolue,

2.11.1 - NOMBRES TRAITES

1 - Structure

Les nombres double précision sont de la forme :

$$A = \underbrace{\alpha\alpha \text{ aaaa aaaa}}_{\text{Mot 1}} \underbrace{+}_{-} \underbrace{\text{aaaaaaaaaa}}_{\text{Mot 2}} \underbrace{+}_{-} = (\alpha, a)$$

Ils sont constitués d'un exposant de 2 chiffres $\alpha\alpha$, et d'une mantisse a de 18 chiffres répartis en 2 tranches : la première formée des 8 premiers chiffres et qui, avec l'exposant, constitue le Mot 1 et la seconde tranche formée des 10 derniers chiffres constituant ainsi le Mot 2.

La mantisse a est normalisée :

$$0,1 \leq a < 1, \text{ cadrée sur la 3ème position du Mot 1.}$$

L'élément central des exposants est :

$$E_0 = \frac{10^2}{2} \approx 50 \text{ de sorte que } \alpha\alpha \text{ varie entre les limites } 01 \text{ et } 99.$$

Le nombre $+1$ s'écrit donc en double précision :

$$\begin{array}{cc} 51 & 1000 & 0000 & + & & 000000 & 0000 & + \\ & \text{Mot 1} & & & & \text{Mot 2} & & \end{array}$$

Tout nombre double précision traité par l'ADP devra donc avoir la forme ci-dessus, en 2 Mots, chacun occupant une mémoire du tambour.

Remarques : Le nombre zéro s'écrit 00 0000 0000 + 000000 0000 +
La mantisse d'un nombre doit être normalisée.

2. Disposition en mémoire

Le Mot 1 et le Mot 2 du nombre N sont chargés dans 2 mémoires d'adresses F et F' telles que :

$$F' - F = D'$$

D' étant la constante de translation ou distance entre les 2 Mots ($0 < D' < 1699$). Cette distance est généralement conservée constante pour tous les nombres traités par un programme de Calcul utilisant l'ADP. Elle doit être chargée en 1732 sous la forme :

$$(1732) = 00. D' .0000 +$$

2.11.2 - PSEUDO - BLOC DE CALCUL

1 - Pseudo Accumulateur : 1700 - 1715 - 1730 contient le premier opérande ou le résultat d'une opération arithmétique sous la forme :

$$\begin{aligned} (1700) &= 00\ 0000\ 00\ \alpha\alpha + \\ (1795) &= 00 \longleftarrow a_1 \longrightarrow \pm \\ (1730) &= \longleftarrow a_2 \longrightarrow \pm \end{aligned}$$

2 - Pseudo Distributeur : 1705 - 1717 - 1731 : contient le second opérande d'une opération arithmétique sous la forme :

$$\begin{aligned} (1705) &= 00\ 0000\ 00\ \beta\beta + \\ (1717) &= 00 \longleftarrow b_1 \longrightarrow \pm \\ (1731) &= \longleftarrow b_2 \longrightarrow \pm \end{aligned}$$

2.11.3 - ETUDE DE L'ARITHMETIQUE

Pour l'analyse des instructions, voir Annexe I.

1 - Encombrement : Le programme occupe 300 mémoires de 1700 à 1999, les zones d'entrée et de sortie standard de la bande 1950 étant réservées. Les mémoires 1848 - 1898 - 1948 et 1950 sont disponibles ainsi que les zones : 1750 - 1756 ; 1768 - 1774 ; 1786 - 1796 et la mémoire 1798.

2 - Présentation : Le programme se présente sous la forme d'un paquet de 50 cartes "6 Mots par carte" (cartes vertes) avec Mot de contrôle en Mot 8 : 10918500n' précédé d'un programme de chargement dans la bande 1950 (cartes jaunes) avec contrôle du numéro des cartes.

3 - Utilisation de l'Arithmétique :

1 - Chargement de D' : La distance positive D' entre les Mots 1 et Mot 2 des nombres traités doit, avant toute opération, être chargée en 1732 sous la forme (1732) = 00 D' 0000 +

En général D' est conservé constant tout au long des calculs : dans le cas contraire, il faudra modifier le contenu de 1732.

2 - A l'entrée de l'ADP :

8001 est quelconque. Son contenu sera détruit au cours de la séquence d'entrée.

(8002) = xx F xxxx, F, en position d'adresse facteur est l'adresse du Mot 1 du nombre traité par la présente opération, qu'il s'agisse d'une opération arithmétique ou d'un transfert. F doit être numérique, de la série 0000-1999 en dehors de l'Arithmétique.

(8003) = instruction de sortie I.

L'entrée dans l'Arithmétique se fait par l'une des 11 entrées de la série 1757 - 1767, chaque entrée étant spécifique d'une opération :

Addition	1757	Soustraction	1758
Addition de valeur absolue	1759	Soustraction de valeur absolue	1762
Multiplication	1761	Division	1760
Transfert vers mémoire générale			1763
Transfert vers le PA de l'opérande			1764
de son opposé			1765
de sa valeur absolue			1766
de l'opposé de sa valeur absolue			1767

Exemple : Multiplication par un nombre dont le Mot 1 est en 1000 et le Mot 2 en 1050. 1732 doit contenir (1732) = 00 0050 0000 +

(8001) = quelconque

(8002) = xx 1000 xxxx

(8003) = instruction de sortie

Entrée dans l'Arithmétique en 1761.

Remarque : Dans le cas d'une opération arithmétique, l'un des opérandes doit être déjà chargé dans le PA, qu'il s'agisse du résultat de l'opération précédente ou bien qu'il y ait été initialement chargé en utilisant l'un des 4 codes de transfert vers le PA.

3 - A la sortie :

• Le résultat apparaît sous la forme 2 Mots en 8002 - 8003

$$(8002-8003) = \gamma\gamma \leftarrow c_1 \rightarrow \leftarrow c_2 \rightarrow \pm$$

(8001) est quelconque sauf dans le cas d'un transfert vers la mémoire générale.

• Le résultat apparaît également dans le PA sous la forme :

$$(1700) = 00 \ 0000 \ 00 \gamma\gamma \ +$$

$$(1715) = 00 \leftarrow c_1 \rightarrow \pm$$

$$(1730) = \leftarrow c_2 \rightarrow \pm$$

ainsi que partiellement en 1746

$$(1746) = \gamma\gamma \leftarrow c_1 \rightarrow \pm$$

• L'instruction de sortie I est prise en 1740 ou dans 8003 cas du transfert vers la mémoire générale.

4 - Arrêts programmés

- 01,0111 8000 : Arrêt overflow : exposant du résultat supérieur à 99
- 01 0222 8000 : Le diviseur est nul
- 01 0333 8000 : La constante D' n'a pas été chargée en 1732 ou bien est nulle

2.11.4 - TEMPS DE CALCUL

Ces temps de calcul sont pris entre l'entrée dans l'arithmétique (1757 - 1767) et la sortie en 1740.

Ces temps sont variables pour 3 raisons :

- les adresses F et F' ne sont pas nécessairement optimisées au regard des instructions de l'arithmétique.
- la structure des nombres traités intervient, le sous-programme opérationnel pouvant être plus ou moins totalement utilisé.
- la structure du résultat de l'opération peut nécessiter ou non un recadrage de mantisse.

Pour toutes ces raisons, nous indiquerons ci-dessous les limites entre lesquelles varient le temps d'opération ainsi que le temps correspondant à une opération normale. Ces temps sont exprimés en millisecondes (1 Mot 650 = 0,096 ms)

1 - Transferts

- Vers la mémoire générale 22 à 32 ms
- Vers le PA : de l'opérande 52 à 62 ms
- de l'opposé de l'opérande... 67 à 77 ms
- de la valeur absolue de
- l'opérande ou de son opposé 57 à 82 ms

2 - Opérations arithmétiques

- Addition : 60 à 230 ms - 180 ms
cas particuliers: 60 à 85 ms
- Soustraction : temps de l'addition + 15 ms
- Addition ou soustraction de valeur absolue : temps d'addition + 25 ms
- Multiplication : 65 à 390 ms - 280 ms
cas particuliers: 65 à 85 ms
- Division : 55 à 550 ms - 400 ms
cas particuliers : 55 à 75 ms

Par cas particulier, il faut comprendre un cas où l'un des opérands est nul.

2.12 - ARITHMETIQUE MIXTE SIMPLE ET DOUBLE PRECISION - PROGRAMME A.S.D.P.

L'Arithmétique Mixte est un programme destiné à traiter les nombres réels, Virgule Flottante Flair, Simple et Double Précision et à effectuer sur ces nombres, en Simple ou en Double Précision, l'une des onze opérations suivantes :

6 opérations arithmétiques : Addition, Soustraction, Addition de valeur absolue, Soustraction de même valeur absolue, Multiplication et Division.

5 transferts : transfert vers la mémoire générale ; transferts vers le PA, du nombre considéré, de son opposé, de sa valeur absolue, de l'opposé de sa valeur absolue.

2.12.1 - NOMBRES TRAITES

1 - Structure :

Les nombres Simple Précision sont de la forme : $A = (\alpha, a) = \alpha\alpha \text{ aaaa aaaa } \overset{+}{-}$

Ils sont constitués d'un exposant de 2 chiffres : $\alpha\alpha$ et d'une mantisse a de 8 chiffres, le tout occupant un seul Mot 650.

Les nombres Double Précision sont de la forme :

$$A = (\alpha, a) = \underbrace{\alpha\alpha \text{ aaaa aaaa}}_{\text{Mot 1}} \overset{+}{-} \underbrace{\text{aaaaaa aaaa}}_{\text{Mot 2}} \overset{+}{-}$$

Ils sont constitués d'un exposant de 2 chiffres : $\alpha\alpha$ et d'une mantisse a de 18 chiffres répartis en 2 tranches : la première formée des 8 premiers chiffres et qui avec l'exposant constitue le Mot 1 et la seconde formée des 10 derniers chiffres constituant le Mot 2.

Dans les 2 cas, la mantisse est normalisée : $0,1 \leq a < 1$ et cadrée sur la 3ème position du Mot 1.

L'élément central des exposants est : $E_0 = \frac{10^2}{2} = 50$ de sorte que α varie entre 01 et 99.

Le nombre $+1$ s'écrit donc :

- en Simple Précision : 51 1000 0000 +
- en Double Précision : 51 1000 0000 + 0000000000 +

Tout nombre simple précision occupe une mémoire du tambour ; tout nombre double précision occupe deux mémoires du tambour.

Remarques : Le nombre zéro s'écrit 00 0000 0000 + ou bien 00 0000 0000 + 00 0000 0000 +

2 - Disposition en mémoire :

Un nombre Simple Précision occupe une mémoire d'adresse F.

Le Mot 1 et le Mot 2 d'un nombre Double Précision occupent 2 mémoires d'adresses F et F' telles que : $F' - F = D'$

D' étant la distance des 2 Mots ($0 < D' < 1649$). Cette distance est généralement conservée constante pour tous les nombres traités par un programme de calculs utilisant l'ADP. Elle doit être chargée sous la forme :

$$(1732) = 00 D' 0000 +.$$

2.12.2 - PSEUDO-BLOC DE CALCUL

1 - Pseudo-Bloc de Calcul - Double Précision

• Pseudo - Accumulateur : 1700 - 1715 - 1730 contenant le premier opérande ou le résultat d'une opération arithmétique sous la forme :

$$(1700) = 00 0000 00 \alpha\alpha +$$

$$(1715) = 00 \leftarrow a_1 \rightarrow \pm$$

$$(1730) = \leftarrow a_2 \rightarrow \pm$$

• Pseudo - Accumulateur : 1705 - 1717 ; 1731 contenant le second opérande sous la forme :

$$(1705) = 00 0000 00 \beta\beta +$$

$$(1717) = 00 \leftarrow b_1 \rightarrow \pm$$

$$(1731) = \leftarrow b_2 \rightarrow \pm$$

2 - Pseudo-Bloc de Calcul - Simple Précision

Il est identique au précédent ; la seule différence est que a_2 et b_2 étant nuls, 1730 - 1731 sont chargés de zéros. Il est important de remarquer que, au contraire de l'AVF, le Pseudo-Bloc de Calcul de l'ASDP est indépendant des registres 8003-2. L'Arithmétique Mixte est essentiellement une Arithmétique Double Précision adaptée au traitement des nombres Simple Précision.

2.12.3 - ETUDE DE L'ARITHMETIQUE

Pour l'analyse des instructions, voir Annexe 2.

1 - Encombrement

Le programme occupe 350 mémoires de 1650 à 1999, les zones d'entrée et de sortie standards de la bande 1950 étant disponibles ainsi que les zones 1750 - 1756 , 1768 - 1774 et les mémoires 1798 - 1898 - 1902 et 1948.

2 - Présentation

Le programme se présente sous la forme d'un paquet de 59 cartes " 6 Mots par carte " (cartes vertes) avec Mot de contrôle en Mot 8 : 10918500 n° , précédé d'un programme de chargement dans la bande 1950 (cartes jaunes) avec contrôle du numéro des cartes et carte remise à zéro du tambour.

3 • Utilisation de l'ASDP en Double Précision

Voit utilisation de l'ADP au § 2.11.3

4 • Utilisation de l'ASDP en Simple Précision

1 • Chargement de 1732

La mémoire 1732, qui dans le cas de l'utilisation en Double Précision contient la distance D n'est pas prise en considération dans le cas d'une utilisation en Simple Précision.

(1732) = quelconque.

2 • A l'entrée de l'Arithmétique

8001 : contenu quelconque détruit au cours de la séquence d'entrée.

(8002) = xx F xxxx + F, en position d'adresse facteur est l'adresse de la mémoire contenant le nombre traité par la présente opération, qu'il s'agisse d'une opération arithmétique ou de transfert ; elle peut être symbolique ou numérique de la série 0000 - 1649 et dans ce dernier cas prise en dehors de la zone de l'Arithmétique.

(8003) = Instruction de sortie I.

L'entrée dans l'Arithmétique se fait dans l'une des 10 entrées de la série 1786 - 1795, chaque entrée étant spécifique d'une opération.

Addition	1786	Soustraction	1787
Addition de valeur absolue	1788	Soustraction de valeur absolue	1791
Multiplication	1790	Division	1789
Transfert vers le PA de l'opérande			1792
de son opposé			1793
de sa valeur absolue			1794
de l'opposé de sa valeur absolue			1795

Le transfert du résultat sur mémoire générale est effectué au moyen du code classique TRG (21).

Exemple : Multiplication par un nombre chargé en 673.

La mémoire 1732 est quelconque.

La séquence d'entrée est la suivante :

(8001) : quelconque

(8002) : xx 0673 xxxx +

(8003) : Instruction de sortie.

Entrée dans l'Arithmétique en 1790.

Remarque : Dans le cas d'une opération arithmétique l'un des opérandes doit être chargé dans le PA, soit comme résultat de l'opération précédente, soit comme résultat d'un transfert vers le PA.

3 • A la sortie

Le résultat apparaît dans 8003.

(8003) = YY ← c →.

et également dans le PA :

$$\begin{aligned}(1700) &= 00\ 0000\ 00\ \gamma\gamma\ + \\(1715) &= 00 \longleftarrow c \longrightarrow \underline{+}\end{aligned}$$

ainsi qu'en 1746 sous la forme : $(1746) = \gamma\gamma \longleftarrow c_1 \longrightarrow \underline{+}$

L'instruction de sortie , I , est prise en 1740.

4 - Arrêts programmés

- 01 0111 8000 Arrêt over flow : exposant du résultat supérieur à 99.
- 01 0222 8000 Diviseur nul.
- 01 0333 8000 En double Précision uniquement, la constante D' est nulle ou bien n'a pas été chargée en 1732.

2.12.4 - TEMPS DE CALCUL

Ces temps de calcul sont pris entre l'entrée dans l'Arithmétique : 1757 • 1767 ou bien 1786 • 1795 et l'instruction de sortie en 1740. Ces durées sont variables pour une même opération :

- Les adresses F et F' ne sont pas nécessairement optimisées vis à vis des instructions de l'Arithmétique.
- La structure des opérandes nécessite une exploitation plus ou moins poussée des sous-programmes opérationnels.
- La structure du résultat peut ou non nécessiter un recadrage (ou deux) de la mantisse.

Pour toutes ces raisons nous indiquerons en millisecondes les limites entre lesquelles varie la durée d'opération, et la durée moyenne d'une opération normale.

1 • Simple Précision

- 1 • Transferts vers le PA ,... de l'opérande 50 • 55 ms
de son opposé 65 à 70 ms
de la valeur absolue du nombre ou de son opposé 55 à 75 ms.

2 • Opérations arithmétiques

- Addition : 55 à 205 ms • 180 ms
cas particuliers : 55 à 70 ms
- Soustraction : majorer le temps d'addition de 10 ms
- Addition ou Soustraction de valeur absolue : majorer le temps d'addition de 5 à 15 ms
- Multiplication : 65 à 225 ms • 200 ms
cas particuliers : 65 à 95 ms
- Division : 65 à 250 ms • 230 ms
cas particuliers : 65 à 75 ms.

2 - Double précision

(voir le § 2.11.4)

Par cas particulier, il faut entendre le cas où l'un des opérandes est nul.

2.13 - ARITHMETIQUE TRIPLE PRECISION - A T P -

L'Arithmétique Triple Précision est un programme destiné à traiter les nombres Virgule Flottante Flair, Triple Précision et à effectuer sur ces nombres l'une des 11 opérations suivantes :

- 6 opérations arithmétiques : Addition, Soustraction, Addition de valeur absolue, Soustraction de valeur absolue, Multiplication et Division.

- 5 transferts : transfert vers la mémoire générale ; transferts vers le P.A d'un nombre, de son opposé, de sa valeur absolue ou de l'opposé de cette valeur absolue.

2.13.1 - NOMBRES TRAITES

1 - Structure

Les nombres triple précision sont de la forme :

$$A = (\alpha, a) = \underbrace{\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha}_{\text{Mot 1}} \pm \underbrace{\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha}_{\text{Mot 2}} \pm \underbrace{\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha}_{\text{Mot 3}}$$

Ils sont constitués d'un exposant de 3 chiffres : $\alpha\alpha\alpha$ et d'une mantisse de 27 chiffres répartis en 3 tranches : la première tranche formée des 7 premiers chiffres constitue avec l'exposant le Mot 1 ; la seconde tranche formée des 10 chiffres suivants constitue le Mot 2 et la troisième formée des 10 derniers chiffres constitue le Mot 3.

La mantisse a est normalisée : $0,1 \leq a < 1$, cadrée sur la 4^{ème} position du Mot 1.

L'élément central des exposants est : $E_0 = \frac{10^3}{2} = 500$ de sorte que α varie entre les limites 001 et 999. Le nombre $+1$ s'écrit donc :

$$\begin{array}{ccc} 501100\ 0000 & + & 0000000000 & + & 0000000000 & + \\ \text{Mot 1} & & \text{Mot 2} & & \text{Mot 3} \end{array}$$

Tout nombre Triple Précision traité par l'A T P devra donc avoir la forme ci-dessus, en 3 Mots, chacun occupant l'emplacement d'une mémoire sur le tambour.

Remarque : Le nombre zéro s'écrit : Mot 1 = Mot 2 = 00 0000 0000 + = Mot 3
La mantisse doit être normalisée.

2 • Disposition en mémoire

Les Mot 1, Mot 2, Mot 3 du nombre N sont chargés dans 3 mémoires d'adresses F, F' et F'' telles que :

$$F' - F = D' \quad \text{et} \quad F'' - F' = D''$$

D' et D'' sont les constantes de translation : ($0 < D'$ ou $D'' < 1548$)

Ces constantes sont généralement gardées fixes pour tous les nombres traités au cours d'un programme de Calcul utilisant l'A T P. Elles doivent être chargées sous la forme :

00 D' 0000 + en 1732
et 00 D'' 0000 + en 1747

2.13.2 • PSEUDO-BLOC DE CALCUL

• Pseudo-Accumulateur : 1700 • 1715 • 1730 • 1745 qui contient le premier opérande ou le résultat d'une opération arithmétique sous la forme :

(1700) = 00 0000 0 $\alpha\alpha\alpha$ +
(1715) = 000 $\leftarrow a_1 \rightarrow \pm$
(1730) = $\leftarrow a_2 \rightarrow \pm$
(1745) = $\leftarrow a_3 \rightarrow \pm$

• Pseudo-Distributeur : 1705 • 1717 • 1731 • 1754 qui contient le second opérande d'une opération arithmétique sous la forme :

(1705) = 000000 0 $\beta\beta\beta$ +
(1717) = 00 0 $\leftarrow b_1 \rightarrow \pm$
(1731) = $\leftarrow b_2 \rightarrow \pm$
(1754) = $\leftarrow b_3 \rightarrow \pm$

2.13.3 • ETUDE DE L'ARITHMETIQUE

Pour l'analyse des instructions, voir annexe 3.

1 • Encombrement

Le programme occupe 450 mémoires de 1550 à 1999, les zones d'entrée et de sortie standard de la bande 1950 étant réservées. Quatre mémoires sont également disponibles : 1945 • 1946 • 1995 • 1996.

2 • Présentation

Le programme se présente sous la forme d'un paquet de 75 cartes " 6 Mots " (cartes vertes) avec Mot de contrôle en Mot 8 = 109185 00 n°, précédé d'un programme de chargement dans la bande 1950 (cartes jaunes) avec contrôle du numéro de carte et carte de remise à zéro du tambour.

3 - Utilisation de l'Arithmétique

1 - Chargement D' et D''

La distance D' entre Mot 1 et Mot 2 et la distance D'' entre Mot 2 et Mot 3 des nombres traités doivent, avant toute opération être chargées respectivement en 1732 et 1747 sous la forme :

$$\begin{aligned}(1732) &= 00 \text{ D}' 0000 + \\(1747) &= 00 \text{ D}'' 0000 +\end{aligned}$$

En général D' et D'' sont conservés constants tout au long des calculs ; dans le cas contraire il faudrait modifier le contenu de 1732 ou 1747.

2 - A l'entrée de l'A T P

(8001) : Contenu quelconque : détruit au cours de la séquence d'entrée.

(8002) = xx F xxxx , F en position d'adresse facteur est l'adresse du Mot 1 du nombre traité par la présente opération, qu'il s'agisse d'une opération arithmétique ou d'un transfert. F ne doit pas être une adresse de l'Arithmétique. F doit être numérique de la série 0000 - 1549.

(8003) = L'instruction de sortie I.

L'entrée dans l'Arithmétique se fait par l'une des 11 entrées de la série 1775 - 1785, chaque entrée étant spécifique d'une opération:

Addition	1775	Soustraction	1776
Addition de valeur absolue	1777	Soustraction de valeur absolue	1780
Multiplication	1779	Division	1778
Transfert vers mémoire générale			1781
Transfert vers le PA	de l'opérande		1782
	de son opposé		1783
	de sa valeur absolue		1784
	de l'opposé de sa valeur absolue		1785

Exemple : multiplication par un nombre dont le Mot 1 est en 0950, le Mot 2 en 0951 et Mot 3 en 0952. On doit avoir (1732) = 00 0001 0000 +
(1747) = 00 0001 0000 +

(8001) = quelconque

(8002) = xx 0950 xxxx

(8003) = Instruction de sortie

Entrée de l'Arithmétique en 1779.

Remarque : Dans le cas d'une opération arithmétique l'un des 2 opérandes doit être déjà chargé dans le PA qu'il s'agisse du résultat d'une opération précédente ou bien qu'il y ait été initialement transféré en utilisant l'un des 4 codes de transfert vers le PA.

3 - A la sortie

• Le résultat apparaît sous la forme 3 Mots en 8001 - 8002 - 8003, sauf dans le cas d'un transfert

du P.A vers mémoire générale.

$$(8003 \cdot 8002 \cdot 8001) = \gamma \gamma \gamma \leftarrow c_1 \rightarrow \leftarrow c_2 \rightarrow \leftarrow c_3 \rightarrow$$

• Il apparaît également dans le P.A sous la forme

$$\begin{aligned} (1700) &= 00 \ 0000 \ 0\gamma\gamma\gamma \ + \\ (1715) &= 000 \leftarrow c_1 \rightarrow \pm \\ (1730) &= \leftarrow c_2 \rightarrow \pm \\ (1745) &= \leftarrow c_3 \rightarrow \pm \end{aligned}$$

ainsi que partiellement en 1746

$$(1746) = \gamma\gamma\gamma \leftarrow c_1 \rightarrow \pm$$

• L'instruction de sortie I est prise en 1740 ou dans 8003 dans le cas du transfert vers la mémoire générale.

4 - Arrêts programmés

- 01 0111 8000 : Arrêt over flow : exposant du résultat supérieur à 999
- 01 0222 8000 : Diviseur nul
- 01 0333 8000 : L'une des constantes D' et D'' n'a pas été chargée ou bien est nulle.

2.13.4 - TEMPS DE CALCUL

Ces temps de calcul sont pris entre l'entrée dans l'arithmétique (1775 - 1785) et la sortie en 1740. Remarquons que ces durées d'opération sont variables pour 3 raisons :

- Les adresses F, F', F'' ne sont pas nécessairement optimisées vis à vis des instructions de l'Arithmétique.
- La structure des nombres traités intervient, puisque le sous-programme opérationnel est alors plus ou moins exploité.
- La structure du résultat de l'opération intervient également, puisqu'elle peut ou non nécessiter un recadrage (ou deux) de la mantisse.

Pour toutes ces raisons, nous indiquerons ci-dessous les limites entre lesquelles varie la durée d'opération ainsi que la durée normale correspondant à une opération normale.

1 • Transferts

- Vers la mémoire générale 64 à 78 ms
- Vers le P.A : de l'opérande 94 à 109 ms
de l'opposé de l'opérande 117 à 130 ms
de la valeur absolue de l'opérande ou de son opposé. 100 à 133 ms

2 • Opérations Arithmétiques

- Addition : 155 à 390 ms • 250 ms
cas particuliers : 155 à 185 ms

- Soustraction : temps d'addition + 23 ms
- Addition ou Soustraction de valeur absolue : temps d'addition + 6 à 25 ms
- Multiplication : 100 à 540 ms • 400 ms
Cas particuliers : 100 à 145 ms
- Division : 100 à 810 ms • 680 ms
Cas particuliers : 100 à 120 ms

Par cas particulier, il faut entendre un cas où l'un des opérandes est nul.

SECTION 2.2

ARITHMETIQUE COMPLEXE

2.21 • ARITHMETIQUE COMPLEXE SIMPLE PRECISION • PROGRAMME A.CxSP •

L'Arithmétique Complexe Simple Précision est un programme destiné à traiter les nombres complexes, Virgule Flottante Simple Précision et à effectuer sur ces nombres l'une des 7 opérations suivantes :

- 4 Opérations arithmétiques : addition, soustraction, Multiplication et division.
- 3 Transferts : transfert vers la mémoire générale ; transferts vers le PACx du nombre ou de son opposé.

2.21.1 - NOMBRES TRAITES

1 • Structure

Tout nombre complexe $Z = (A, B)$ se présente sous la forme :

Mot 1 = $\alpha\alpha \leftarrow a \rightarrow \pm$: partie réelle

Mot 2 = $\beta\beta \leftarrow b \rightarrow \pm$: partie imaginaire

parties réelle et imaginaire étant des nombres réels Virgule Flottante Simple Précision. Chacune de ces 2 parties occupant 1 Mot 650 est formée d'un exposant de 2 chiffres et d'une mantisse de 8 chiffres (Cf. Simple Précision).

Remarques : Chacun de ces 2 Mots peut porter un signe différent. Un nombre réel est caractérisé par un Mot 2 nul et un nombre imaginaire par un Mot 1 nul.

2 • Disposition en mémoire

Les Mots 1 et 2 du nombre Z sont chargés dans 2 mémoires d'adresses F et G telles que :

$$G - F = \Delta$$

Δ étant la constante de translation ou distance entre partie réelle et partie imaginaire. $0 < \Delta < 1650$.

Cette distance est généralement gardée constante pour tous les nombres complexes traités au cours d'un programme de calcul utilisant l'Arithmétique Complexe. Elle doit être chargée sous la forme :

00 Δ 0000 + en 1732

2.21.2 • PSEUDO-BLOC DE CALCUL

Le Pseudo Accumulateur Complexe PACx : 1650 - 1655 contient le premier opérande ou le résultat d'une opération arithmétique sous la forme :

(1650) = $\alpha\alpha \leftarrow a \rightarrow$ Mot 1 : partie réelle

(1655) = $\beta\beta \leftarrow b \rightarrow$ Mot 2 : partie imaginaire

Le Pseudo Distributeur Complexe PDCx : 1662 - 1667 contient le second opérande d'une opération arithmétique sous la forme :

$$\begin{aligned}(1662) &= \alpha' \alpha'' \rightarrow a' \longrightarrow \text{Mot 1 : partie réelle} \\ (1667) &= \beta' \beta'' \rightarrow b' \longrightarrow \text{Mot 2 : partie imaginaire}\end{aligned}$$

2.21.3 - ETUDE DE L'ARITHMETIQUE

Pour l'analyse des instructions, voir Annexe 4 .

1 - Encombrement

Le programme occupe 350 mémoires réparties en 2 zones :

1650 - 1799 : Arithmétique Complexe
1800 - 1999 : A. V. F.

Les zones d'entrée et de sortie Standards de la bande 1950 sont disponibles ainsi que 12 mémoires générales dans la zone de l'Arithmétique Complexe :

1651 - 1697 - 1698
1701 - 1702 - 1744 - 1746 - 1747 - 1748
1794 - 1797 et 1798.

2 - Présentation

Le programme complet (Arithmétique Complexe et A.V.F.) se présente sous la forme d'un paquet de 59 cartes " 6 Mots par carte " (cartes vertes) avec Mot de contrôle en Mot 8 : 109 185 00 n°, précédé d'un programme de chargement dans la bande 1950 (cartes jaunes) avec contrôle de la numérotation et carte remise à zéros du tambour.

3 - Utilisation de l'Arithmétique

1 - Chargement de Δ

La distance Δ entre Mot 1 et Mot 2, partie réelle et partie imaginaire des nombres traités doit, avant toute opération, être chargée en 1732 sous la forme :

$$(1732) = 00 \Delta 0000 +$$

En général Δ est conservé constant tout au long des calculs ; dans le cas contraire le contenu de 1732 devrait être modifié.

2 - A l'entrée de l'Arithmétique

8001 : contenu quelconque, détruit au cours de la séquence d'entrée .

(8002) = xx F xxxx + , F en position d'adresse facteur est l'adresse du Mot 1 traité par la présente opération, qu'il s'agisse d'une opération arithmétique ou d'un transfert. F doit être une adresse numérique de la série 0000 - 1999 , prise en dehors de l'Arithmétique.

(8003) = Instruction de sortie I

L'entrée dans l'Arithmétique se fait dans l'une des 7 entrées de la série 1750 - 1756, chaque entrée étant spécifique de l'opération :

Addition :	1750	Soustraction :	1751
Multiplication :	1753	Division :	1752
Transfert sur mémoire générale :			1754
Transfert vers le PACx, du nombre :			1755
	de son opposé :		1756

Exemple : Multiplication par un nombre dont le Mot 1 est chargé en 1250 et le Mot 2 en 1260. On doit avoir en 1732 : 00 0010 0000 + et la séquence d'entrée est :

(8001) : quelconque
 (8002) : xx 1250 xxxx +
 (8003) : Instruction de sortie.
 Entrée dans l'Arithmétique : 1753.

Remarque : Dans le cas d'une opération arithmétique l'un des 2 opérandes doit être initialement chargé dans le PAC, soit comme résultat d'une opération antérieure, soit comme résultat d'un transfert dans le PACx.

3 - A la sortie :

Le résultat apparait dans le PACx sous la forme :

(1650) = $\gamma\gamma \leftarrow c \rightarrow \pm$ Mot 1 : partie réelle
 (1655) = $\delta\delta \leftarrow d \rightarrow \pm$ Mot 2 : partie imaginaire

Les contenus de 8001 - 2 - 3 sont variables suivant l'opération dont est issu le résultat.

L'instruction de sortie I est prise en 1740 (ou 1677 dans le cas de la division).

4 - Arrêts programmés

- 01 0111 8000 - Arrêt over flow - l'exposant du résultat est supérieur à 99
- 01 0222 8000 - Diviseur nul
- 01 0333 8000 - La distance Δ n'a pas été chargée en 1732 ou bien est nulle. Cet arrêt est spécifique de l'Arithmétique complexe.

2.21.4 - TEMPS DE CALCUL

Ces temps de calcul sont pris entre l'entrée dans l'Arithmétique (1750 - 1756) et la sortie en 1740 (ou 1677). Ces durées sont variables pour une même opération :

- les adresses F et G ne sont pas nécessairement optimisées vis à vis des instructions de l'Arithmétique.
- la structure des nombres traités nécessite une exploitation plus ou moins poussée du sous-programme opérationnel.

1 - Transferts :

Vers la mémoire générale : 24 à 33 ms
 Vers le PACx du nombre : 27 à 36 ms
 de son opposé : 27 à 36 ms

2 - Opérations Arithmétiques

Addition : de 175 à 190 ms
 Soustraction : de 200 à 220 ms
 Multiplication : de 440 à 460 ms
 Division : de 840 à 875 ms

Remarque :

Le programme de l'Arithmétique Complexe faisant appel à l'A V F permet également de traiter tout problème en simple Précision sur des nombres réels puisque l'A V F est incluse à l'Arithmétique Complexe.

2.22 • ARITHMETIQUE COMPLEXE DOUBLE PRECISION • PROGRAMME : A Cx D P •

L'Arithmétique Complexe Double Précision est un programme destiné à traiter les nombres Complexes, Virgule Flottante Double Précision et à effectuer sur ces nombres l'une des 7 opérations suivantes :

- 4 opérations arithmétiques : Addition, Soustraction, Multiplication, Division.
- 3 Transferts : transfert vers la mémoire générale ; transferts vers le PACx d'un nombre ou de son opposé.

2.22.1 • NOMBRES TRAITES

1 - Structure

Tout nombre complexe $Z = (A, B)$ double précision se présente sous la forme :

$$\begin{array}{lcl} \text{Mot 1} = \alpha \alpha \leftarrow a_1 \rightarrow \pm & & \\ \text{Mot 2} = \leftarrow a_2 \rightarrow \pm & & \\ & & \left. \vphantom{\begin{array}{l} \text{Mot 1} \\ \text{Mot 2} \end{array}} \right\} \text{partie réelle} \\ \\ \text{Mot 3} = \beta \beta \leftarrow b_1 \rightarrow \pm & & \\ \text{Mot 4} = \leftarrow b_2 \rightarrow \pm & & \\ & & \left. \vphantom{\begin{array}{l} \text{Mot 3} \\ \text{Mot 4} \end{array}} \right\} \text{partie imaginaire} \end{array}$$

parties réelle et imaginaire étant des nombres réels, Virgule Flottante, Double Précision. Chacune de ces 2 parties occupe 2 Mots 650, le premier constitué par l'exposant de 2 chiffres et les 8 premiers chiffres de la mantisse, le second formé des 10 derniers chiffres de la mantisse.

Remarques : Les Mots 1 et 2 ont même signe ; les Mots 3 et 4 ont même signe ; par contre partie réelle et imaginaire peuvent avoir des signes différents . Pour un nombre réel, les Mots 3 et 4 sont nuls ; pour un nombre imaginaire pur, les Mots 1 et 2 sont nuls.

2 - Disposition en mémoire

Les 4 Mots du nombre Z sont chargés dans 4 mémoires d'adresses F et F', G et G' telles que :

$$\begin{array}{lcl} G - F = G' - F' = \Delta & 0 < \Delta < 1515 \\ \text{et} & F' - F = G' - G = D' & 0 < D' < 1515 \end{array}$$

Δ étant la distance entre partie réelle et imaginaire, D' la distance entre les 2 Mots de chacune de ces parties. Ces distances sont généralement gardées constantes pour tous les nombres complexes traités au cours d'un programme de calcul utilisant l'Arithmétique Complexe. Elles doivent y être chargées sous la forme :

00 Δ 0000 + en 1541
00 D' 0000 + en 1581

2.22.2 - PSEUDO-BLOC DE CALCUL

Le Pseudo-Accumulateur Complexe : PACx : 1650- 1665 et 1655 - 1670 contient le premier opérande ou le résultat d'une opération arithmétique sous la forme :

$$\begin{array}{lcl} (1650) = \alpha\alpha \leftarrow a_1 \rightarrow \pm \text{ Mot 1} \\ (1665) = \leftarrow a_2 \rightarrow \pm \text{ Mot 2} \end{array} \left. \vphantom{\begin{array}{l} (1650) \\ (1665) \end{array}} \right\} \text{Partie réelle}$$

$$\begin{array}{lcl} (1655) = \beta\beta \leftarrow b_1 \rightarrow \pm \text{ Mot 3} \\ (1670) = \leftarrow b_2 \rightarrow \pm \text{ Mot 4} \end{array} \left. \vphantom{\begin{array}{l} (1655) \\ (1670) \end{array}} \right\} \text{Partie imaginaire.}$$

Le Pseudo-Distributeur Complexe : PDCx : 1662 - 1677 et 1667 - 1682 contient le second opérande d'une opération arithmétique sous la forme :

$$\begin{array}{lcl} (1662) = \alpha' \alpha' \leftarrow a'_1 \rightarrow \pm \text{ Mot 1} \\ (1677) = \leftarrow a'_2 \rightarrow \pm \text{ Mot 2} \end{array} \left. \vphantom{\begin{array}{l} (1662) \\ (1677) \end{array}} \right\} \text{Partie réelle}$$

$$\begin{array}{lcl} (1667) = \beta' \beta' \leftarrow b'_1 \rightarrow \pm \text{ Mot 3} \\ (1682) = \leftarrow b'_2 \rightarrow \pm \text{ Mot 4} \end{array} \left. \vphantom{\begin{array}{l} (1667) \\ (1682) \end{array}} \right\} \text{Partie imaginaire}$$

2.22.3 - ETUDE DE L'ARITHMETIQUE

Pour l'analyse des instructions, voir Annexe 5.

1 - Encombrement

Le programme complet occupe 460 mémoires réparties en 2 zones : 1540 - 1699 : Arithmétique

Complexe et 1700 - 1999 : A.D.P., les zones d'entrée et de sortie standards de la bande 1950 étant disponibles ainsi que la mémoire 1598 - 1648 et 1698.

2 - Présentation

Le programme complet se présente sous la forme d'un paquet de 77 cartes " 6 Mots par carte " (cartes vertes) avec Mot de contrôle en Mot 8 : 109 185 00 n° , précédé d'un programme de chargement 6 Mots dans la bande 1950 avec contrôle de la numérotation et carte de remise à zéro du tambour.

3 - Utilisation de l'Arithmétique

1 - Chargement de Δ et D'

La distance Δ entre partie réelle et partie imaginaire, et la distance D' entre les 2 Mots de chacune de ces parties des nombres traités doivent, avant toute opération, être chargées en 1541 et 1581 sous la forme :

$$\begin{aligned}(1541) &= 00 \Delta 0000 + \\ (1581) &= 00 D' 0000 +\end{aligned}$$

En général Δ et D' sont conservés constants tout au long des calculs : dans le cas contraire, les contenus de 1581 ou 1541 devraient être modifiés.

2 - A l'entrée de l'Arithmétique

8001 : contenu quelconque, détruit au cours de la séquence d'entrée.

(8002) = xx F xxxx + , F en position d'adresse facteur étant l'adresse du Mot 1 du nombre traité par la présente opération, qu'il s'agisse d'une opération arithmétique ou d'un transfert. F doit être une adresse numérique de la série 0000 - 1999 , prise en dehors de l'Arithmétique.

(8003) = Instruction de sortie I.

L'entrée dans l'arithmétique se fait par l'une des 7 entrées de la série 1768 - 1774, chaque entrée étant spécifique d'une opération :

Addition :	1768	Soustraction :	1769
Multiplication :	1771	Division :	1770
Transfert sur mémoire générale :			1772
Transfert vers le PAC du nombre :			1773
		de son opposé :	1774

Exemple : Multiplication par un nombre Complexe, Double Précision chargé :

Mot 1 et Mot 2 1100 et 1101
Mot 3 et Mot 4 1120 et 1121

On doit charger en 1541 : 00 0020 0000 +
1581 : 00 0001 0000 +

et la séquence d'entrée est : (8001) : quelconque
(8002) : xx 1100 xxxx +
(8003) : instruction de sortie
Entrée dans l'Arithmétique : 1771

Remarque : Dans le cas d'une opération arithmétique l'un des 2 opérandes doit être initialement chargé dans le PACx, soit comme résultat d'une opération antérieure, soit comme résultat d'un transfert dans le PACx.

3 - A la sortie

Le résultat apparaît dans le PACx sous la forme :

$$\begin{array}{ll}
 (1650) = \overleftarrow{\gamma\gamma} \overleftarrow{c_1} \overrightarrow{\pm} & \text{Mot 1} \\
 (1665) = \overleftarrow{} \overleftarrow{c_2} \overrightarrow{\pm} & \text{Mot 2}
 \end{array}
 \left. \vphantom{\begin{array}{l} (1650) \\ (1665) \end{array}} \right\} \text{Partie réelle}$$

$$\begin{array}{ll}
 (1655) = \overleftarrow{\delta\delta} \overleftarrow{d_1} \overrightarrow{\pm} & \text{Mot 3} \\
 (1670) = \overleftarrow{} \overleftarrow{d_2} \overrightarrow{\pm} & \text{Mot 4}
 \end{array}
 \left. \vphantom{\begin{array}{l} (1655) \\ (1670) \end{array}} \right\} \text{Partie imaginaire}$$

Les contenus de 8001 - 8002 - 8003 sont variables suivant l'opération dont le résultat est issu. L'instruction de sortie est prise en 1625 (ou 1584 dans le cas de la division).

4 - Arrêts programmés

- 01 0111 8000 : Arrêt overflow : l'exposant du résultat est supérieur à 99.
- 01 0222 8000 : Diviseur nul.
- 01 0333 8000 : L'une au moins des 2 constantes Δ et D' n'a pas été chargée ou bien est nulle.

2.22.4 - TEMPS DE CALCUL

Ces temps de calcul sont pris entre l'entrée dans l'Arithmétique (1768 - 1774) et la sortie en 1625 (ou 1584). Ces durées sont variables pour une même opération :

- les adresses F, F', G et G' ne sont pas nécessairement optimisées vis à vis des instructions de l'Arithmétique.
- la structure des opérandes nécessite une exploitation plus ou moins poussée des sous-programmes opérationnels.
- la structure du résultat peut ou non nécessiter un recadrage de mantisse.

Pour toutes ces raisons, nous indiquerons en millisecondes les limites entre lesquelles varie normalement la durée d'opération, calculées à partir des temps moyens des opérations virgule flottante de l'ADP.

1 - Transferts

Vers la mémoire générale : 56 à 75 ms

Vers le PACx, du nombre ou de son opposé : 80 à 100 ms.

2 - Opérations arithmétiques

Addition : 520 à 660 ms

Soustraction : 545 à 685 ms

Multiplication : 1670 à 2340 ms

Division : 3340 à 4250 ms

Remarque: Le programme de l'Arithmétique Complexe faisant appel à l'ADP permet également de traiter tout problème en double précision sur des nombres réels double précision.

2.23 • ARITHMETIQUE COMPLEXE MIXTE SIMPLE ET DOUBLE PRECISION • PROGRAMME : ACxSDP

L'Arithmétique Complexe Mixte, Simple et Double Précision est un programme destiné à traiter les nombres complexes, Virgule Flottante, Simple et Double Précision et à effectuer sur ces nombres l'une des 7 opérations suivantes :

- 4 opérations arithmétiques : Addition, Soustraction, Multiplication, Division.
- 3 transferts : transfert vers la mémoire générale ; transfert vers le PACx du nombre ou de son opposé.

2.23.1 • NOMBRES TRAITES

1 • Structure en Simple Précision

Tout nombre complexe $Z = (A, B)$ simple précision se présente sous la forme :

$$\begin{array}{lcl} \text{Mot 1} & = & \alpha\alpha \leftarrow a \rightarrow \pm \\ \text{Mot 2} & = & \beta\beta \leftarrow b \rightarrow \pm \end{array} \quad \begin{array}{l} \text{: partie réelle} \\ \text{: partie imaginaire,} \end{array}$$

parties réelle et imaginaire étant des nombres réels Virgule Flottante Simple Précision. Chacune de ces parties occupant 1 Mot 650 est formée d'un exposant de 2 chiffres et d'une mantisse de 8 chiffres.

2 • Structure en Double Précision

Tout nombre complexe $Z = (A, B)$ se présente en double précision sous la forme :

$$\begin{array}{lcl} \text{Mot 1} & = & \alpha\alpha \leftarrow a_1 \rightarrow \pm \\ \text{Mot 2} & = & \leftarrow a_2 \rightarrow \pm \end{array} \quad \left. \vphantom{\begin{array}{l} \text{Mot 1} \\ \text{Mot 2} \end{array}} \right\} \text{partie réelle}$$

$$\begin{array}{lcl} \text{Mot 3} & = & \beta\beta \leftarrow b_1 \rightarrow \pm \\ \text{Mot 4} & = & \leftarrow b_2 \rightarrow \pm \end{array} \quad \left. \vphantom{\begin{array}{l} \text{Mot 3} \\ \text{Mot 4} \end{array}} \right\} \text{partie imaginaire}$$

parties réelle et imaginaire étant des nombres réels, Virgule Flottante, Double Précision. Chacune de ces parties occupe 2 Mots 650, le premier Mot étant formé de l'exposant de 2 chiffres et des 8 premiers chiffres de la mantisse, le second, des 10 derniers chiffres de la mantisse.

3 • Disposition en mémoire

Cas de la Double Précision

Les 4 Mots du nombre Z sont chargés dans 4 mémoires d'adresses F et F' , G et G' telles que :

$$\begin{aligned} G - F &= G' - F' = \Delta & 0 < \Delta < 1399 \\ F' - F &= G' - G = D' & 0 < D' < 1399 \end{aligned}$$

Δ étant la distance entre partie réelle et imaginaire et D' la distance entre les 2 Mots de chacune de ces parties.

Cas de la Simple Précision

Les 2 Mots du nombre Z sont chargés dans 2 mémoires d'adresses F et G telles que :

$$G - F = \Delta \quad 0 < \Delta < 1399$$

La distance D' , dans ce cas, est nulle et on doit charger $D' = 0000$ dans le cas de la Simple Précision.

Ces distances sont généralement conservées constantes pour tous les nombres complexes traités au cours d'un programme de calcul utilisant l'Arithmétique Complexe Mixte. Elles doivent être chargées sous la forme :

$$\begin{aligned} 00 \Delta 0000 + & \quad \text{en } 1431 \\ 00 D' 0000 + & \quad \text{en } 1441 \end{aligned}$$

En Simple Précision on doit avoir $1441 = 00 \ 0000 \ 0000 +$

2.23.2 - PSEUDO - BLOC DE CALCUL

1 - Pseudo - Bloc de Calcul en Double Précision

Le Pseudo-Accumulateur Complexe : PACx : 1550 , 1565 et 1555 - 1570 contient le premier opérande ou le résultat d'une opération arithmétique sous la forme :

$$\begin{aligned} (1550) &= \alpha\alpha \leftarrow a_1 \rightarrow \pm & \text{Mot 1} \\ (1565) &= \leftarrow a_2 \rightarrow \pm & \text{Mot 2} \end{aligned} \left. \vphantom{\begin{aligned} (1550) \\ (1565) \end{aligned}} \right\} \text{partie réelle}$$

$$\begin{aligned} (1555) &= \beta\beta \leftarrow b_1 \rightarrow \pm & \text{Mot 3} \\ (1570) &= \leftarrow b_2 \rightarrow \pm & \text{Mot 4} \end{aligned} \left. \vphantom{\begin{aligned} (1555) \\ (1570) \end{aligned}} \right\} \text{partie imaginaire}$$

Le Pseudo-Distributeur Complexe PDCx : 1562 - 1577 et 1567 - 1582 contient le second opérande d'une opération arithmétique sous la forme :

$$\begin{aligned} (1562) &= \alpha' \alpha' \leftarrow a'_1 \rightarrow \pm & \text{Mot 1} \\ (1577) &= \leftarrow a'_2 \rightarrow \pm & \text{Mot 2} \end{aligned} \left. \vphantom{\begin{aligned} (1562) \\ (1577) \end{aligned}} \right\} \text{partie réelle}$$

$$\begin{aligned} (1567) &= \beta' \beta' \leftarrow b'_1 \rightarrow \pm & \text{Mot 3} \\ (1582) &= \leftarrow b'_2 \rightarrow \pm & \text{Mot 4} \end{aligned} \left. \vphantom{\begin{aligned} (1567) \\ (1582) \end{aligned}} \right\} \text{partie imaginaire}$$

2 - Pseudo - Bloc de Calcul en Simple Précision

Il est formé des mêmes mémoires que précédemment, mais comme a_2 , b_2 , a'_2 , b'_2 sont nuls, les mémoires 1565 , 1570 et 1577, 1582 sont chargées de zéros.

2.23.3 • ETUDE DE L'ARITHMETIQUE

Pour l'analyse des instructions, voir Annexe 6.

1 • Encombrement

Le programme occupe 600 mémoires réparties en 2 zones :

1400 • 1649 : Arithmétique Complexe
1649 • 1999 : A.S.D.P.

les zones d'entrée et de sortie standards de la bande 1950 étant réservées et la mémoire 1601 étant libre.

2 • Présentation

Le programme complet se présente sous la forme d'un paquet de 100 cartes " 6 Mots par carte " (cartes vertes) avec Mot de contrôle en Mot 8 : 109 185 0 n° précédé d'un programme de chargement dans la bande 1950 (cartes jaunes) avec contrôle de la numérotation et avec carte de remise à zéro du tambour.

3 • Utilisation de l'Arithmétique en Double Précision

1 • Chargement de Δ et D'

La distance Δ entre partie réelle et partie imaginaire et la distance D' entre les 2 Mots de chacune de ces parties des nombres traités doivent, avant toute opération, être chargées en 1431 et 1441 sous la forme :

$$\begin{aligned}(1431) &= 00 \Delta 0000 + \\(1441) &= 00 D' 0000 +\end{aligned}$$

En général Δ et D' sont conservés constants tout au long des calculs ; dans le cas contraire les contenus de 1431 et 1441 devraient être modifiés.

2 • A l'entrée dans l'Arithmétique

3 • A la sortie

(Voir Arithmétique Complexe Double Précision , § 2.22.3)

L'instruction de sortie I est prise en 1625 (ou 1531 dans le cas de la division)

4 • Utilisation de l'Arithmétique en Simple Précision

1 • Chargement de Δ et D'

La distance Δ entre partie réelle et partie imaginaire des nombres traités doit, avant toute opération, être chargée en (1431) sous la forme :

$$(1431) = 00 \Delta 0000 +$$

Le contenu de la mémoire (1441), dans le cas de l'utilisation en Simple Précision, doit être chargé de zéros avant toute opération.

$$(1441) = 00 0000 0000 +$$

En général, Δ est conservé constant tout au long des calculs ; dans le cas contraire, le contenu de (1431) devrait être modifié.

2 - A l'entrée dans l'Arithmétique

Voir Arithmétique Complexe Simple Précision § 2.21.3 , 1565, 1570 et 1577, 1582 étant chargés de zéros.

3 - A la sortie

Le résultat apparait dans le PACx sous la forme :

$$\begin{aligned} (1550) &= \gamma\gamma \longleftrightarrow c \longrightarrow \pm \text{ Mot 1 : partie réelle} \\ (1555) &= \delta\delta \longleftrightarrow d \longrightarrow \pm \text{ Mot 2 : partie imaginaire} \end{aligned}$$

Les mémoires 1565 et 1570 étant chargées de zéros.

Les contenus de 8001 - 2 - 3 sont variables suivant l'opération dont est issu le résultat.

L'instruction de sortie I est prise en 1625 (en 1531 dans le cas de la division).

4 - Arrêts programmés

- 01 0111 8000 : Arrêt over flow : exposant du résultat supérieur à 99
- 01 0222 8000 : Diviseur nul
- 01 0333 8000 : Δ n'est pas chargé en 1431 ou bien est nul.

Remarque : L'arrêt sur D' nul n'existe plus ; D' nul signifie que l'opération se fait en Simple Précision.

2.23.4 - TEMPS DE CALCUL

Ces temps de calcul sont pris entre l'entrée dans l'Arithmétique : 1750 - 1756 et 1768 - 1774 et l'instruction de sortie en 1625 (ou 1531) . Ces durées sont variables pour une même opération :

- les adresses F, F', G et G' ne sont pas nécessairement optimisées vis à vis des instructions de l'Arithmétique .
- la structure des opérandes nécessite une exploitation plus ou moins poussée des différents sous-programmes opérationnels.
- la structure du résultat peut ou non nécessiter un recadrage (ou deux) de mantisse.

Pour toutes ces raisons, nous indiquerons en millisecondes les limites entre lesquelles varie normalement la durée d'opération :

Simple Précision

1 - Transferts

- vers la mémoire générale 24 à 33 ms
- vers le PACx : du nombre
ou de son opposé 81 à 100 ms

2 - Opérations Arithmétiques

Addition :	485 à 625 ms
Soustraction :	520 à 660 ms
Multiplication :	1340 à 1720 ms
Division :	2530 à 3250 ms

Double précision

1 - Transferts :

vers la mémoire générale : 56 à 75 ms
vers le PACx : du nombre
ou de son opposé : 80 à 100 ms

2 - Opérations arithmétiques

Addition :	480 à 660 ms
Soustraction :	505 à 685 ms
Multiplication :	1540 à 2150 ms
Division :	2860 à 4380 ms

Remarque : Le programme de l'Arithmétique Complexe Mixte comportant le programme de l'Arithmétique réelle mixte, il est possible d'utiliser ce programme comme l'Arithmétique réelle mixte.

SECTION 2.3

EXEMPLES D'UTILISATION

2.31 - CONSTRUCTION D'UNE TABLE DE RACINE CARREE EN DOUBLE PRECISION

On désire construire une table de racine carrée et d'inverse de racine carrée pour les nombres réels compris entre A_1 et A_n ($A_n > A_1$) au pas h , en double précision.

2.31.1 - METHODE UTILISEE

On utilise la méthode itérative de Newton : $x_{i,p+1} = 0,5 \left(x_{i,p} + \frac{A_i}{x_{i,p}} \right)$

$x_{i,p}$ et $x_{i,p+1}$ étant 2 approximations successives de la racine carrée de A_i . Comme valeur initiale $x_{i,0}$ on prend la racine précédemment calculée, $\sqrt{A_{i-1}}$. La valeur $x_{1,0}$ utilisée pour le calcul de la racine de A_1 sera donnée par la carte entrée en même temps que les caractéristiques de la table à construire.

2.31.2 - ORGANIGRAMME

(Voir page suivante)

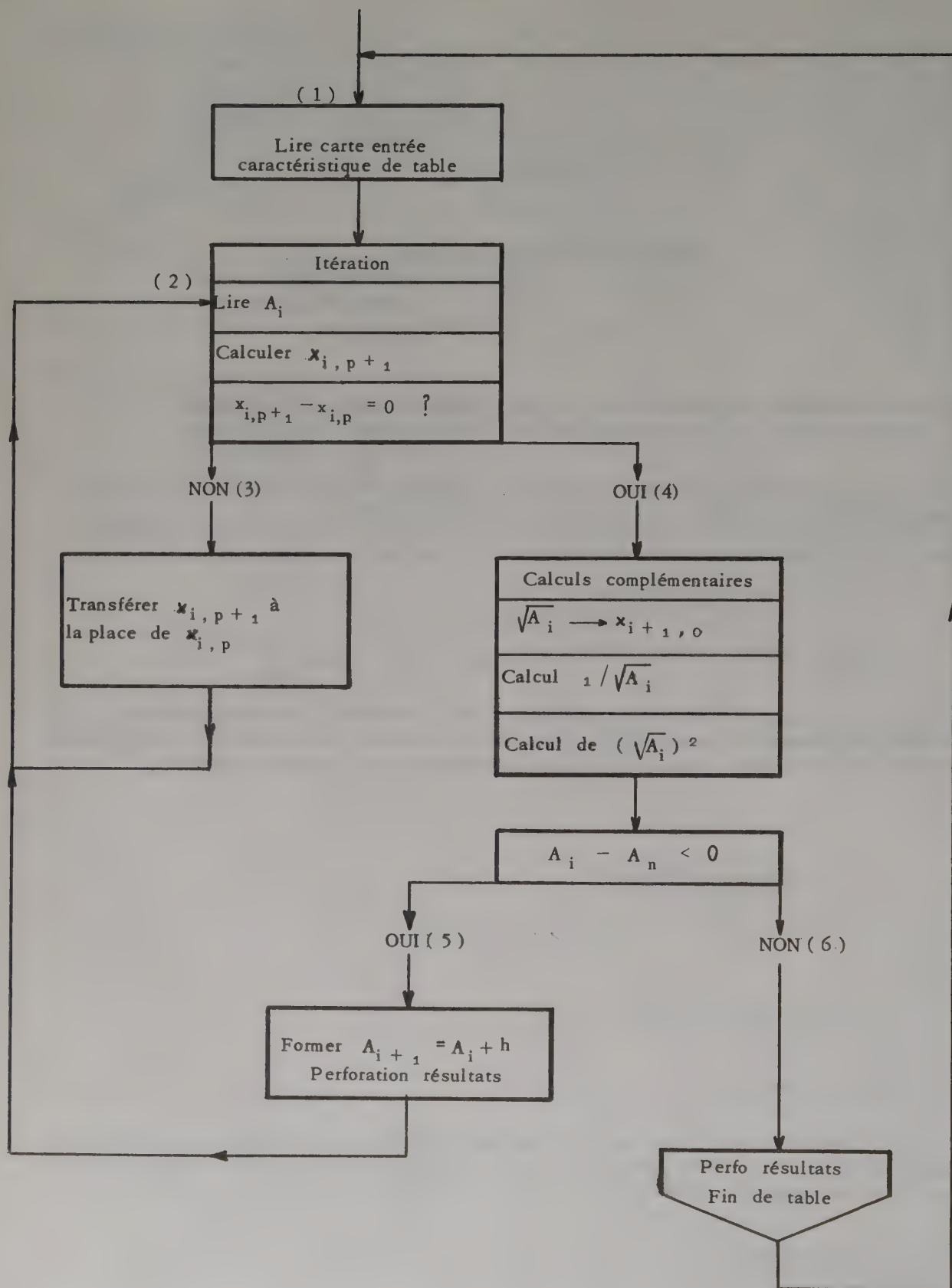
2.31.3 - UTILISATION

1 - Caractéristiques du programme

Il se présente sous la forme 1 Mot par carte (cartes bleues) avec carte départ calcul automatique. Il occupe la zone 1550 - 1699 : 1550 - 1560 réservées aux constantes numériques et 1561 - 1699 au programme proprement dit avec 77 mémoires de livres.

2 - Format de la "carte entrée"

La zone d'entrée est celle de la bande 1950 ; la carte est non chargement et les données carac-



téristiques de la table à construire sont exprimées en VF , Double Précision :

- Mot 1 et 2 : A_1 premier terme de la table
- Mot 3 et 4 : h pas de la table
- Mot 5 et 6 : A_n dernier terme de la table
- Mot 7 et 8 : $x_{1,0}$ première approximation de $\sqrt{A_1}$

3 - Mode d'emploi

Placer le programme de l'ADP (cartes vertes) précédé de sa carte remise à zéro et de son programme de chargement ; placer ensuite le programme de calcul de racine carrée (cartes bleues) et en fin, la ou les cartes caractéristiques des tables à construire.

Inscrire au pupitre 70 0004 xxxx + pour la remise à zéro. Appuyer sur REST, GENE, DEPART PROGRAMME et DEPART LECTURE.

Après la remise à zéro, inscrire 70 1951 xxxx + et appuyer sur DEPART PROGRAMME.

Après chargement des programmes, le départ se fait automatiquement en 1600.

Lorsque tous les calculs ont été faits, le programme s'arrête sur l'instruction de lecture :

70 1950 1650 +

2.31.4 - RESULTATS

1 - Format des cartes sorties

La sortie se fait dans la bande 1950, donc en 1977 - 1986 :

- Mot 1 et 2 : A_i en Double précision
- Mot 3 et 4 : $\sqrt{A_i}$ en Double précision
- Mot 5 et 6 : $1/\sqrt{A_i}$
- Mot 7 et 8 : Vérification : $(\sqrt{A_i})^2$

2 - Temps de calcul

Le temps global de calcul : calcul de racine carrée, de l'inverse, de la vérification et perforation

est de : 7,4 secondes par A_i

3 - Précision sur le résultat

Les résultats, $\sqrt{A_i}$ et $1/\sqrt{A_i}$ sont connus avec 18 chiffres significatifs et la vérification effectuée permet de se faire une idée de la précision du résultat.

$$\text{On a : } \frac{\Delta(\sqrt{A_i})}{(\sqrt{A_i})} \leq \frac{1}{2} \cdot \frac{\Delta[(\sqrt{A_i})^2]}{[(\sqrt{A_i})^2]}$$

L'erreur maximum décelée sur les vérifications est de +2 unités du dernier ordre ; or comme les A_i varient de 1,1 à 3,1, on peut écrire :

$$\frac{\Delta(\sqrt{A_i})}{(\sqrt{A_i})} < \frac{1}{2} ; \frac{2}{1,1} = 1 \text{ unité du dernier ordre.}$$

Donc, puisque $\sqrt{A_i}$ varie en 1 et 1,8, on peut affirmer que seul le 18ème chiffre significatif est légèrement erroné et que l'erreur reste inférieure en valeur absolue à 2 unités du dernier ordre.

La comparaison avec un calcul fait en triple précision confirme ce résultat.

Par exemple on calcule :

$$\sqrt{3} = 1,732 \dots \quad 7730 \text{ alors que, en triple précision on trouve :}$$

$$1,732 \dots \quad 7729 \text{ 35 ...}$$

1 • Mémoires à contenu constant

1550 - 1551	:	5050000000 0000000000 + Multiplicateur + 0,5
1552 - 1553	:	5110000000 0000000000 + Unité + 1
1603	:	00 0001 0000 + Distance entre les 2 Mots des nombres traités
1562	:	00 1955 0000 + Adresse du Mot 1 de A_n
1564	:	00 1552 0000 + Adresse du Mot 1 de l'unité
1571	:	00 1979 0000 + Adresse du Mot 1 de la racine calculée transférée sur zone de perforation
1581	:	00 1981 0000 + Adresse de transfert du Mot 1 de l'inverse de la racine
1588	:	00 1951 0000 + Adresse du Mot 1 de A_1 sur zone d'entrée
1599	:	00 1957 0000 + Adresse du Mot 1 de $x_{1,0}$
1610	:	00 1550 0000 + Adresse du Mot 1 du multiplicateur + 0,5
1632	:	00 1983 0000 + Adresse de transfert du Mot 1 du carré de $\sqrt{A_i}$
1649	:	00 1977 0000 + Adresse de transfert du Mot 1 de A_i
1699	:	00 1953 0000 + Adresse du Mot 1 du pas h
1953 - 1954	:	le pas h de la table en double précision
1955 - 1956	:	A_n , dernier terme de la table en double précision.

2 - Mémoires de travail à contenus variables

Ce sont essentiellement les mémoires des zones d'entrée et sortie de la bande 1950.

- 1951 , 1952 : A_i en double précision en début de calcul et A_i par la suite.
- 1957 - 1958 : $x_{i,0}$ en début de calcul, $x_{i,p}$ en cours d'itération et $x_{i+1,0} = \sqrt{A_i}$ pour la première itération dans le calcul de la racine de A_{i+1}
- 1977 - 1978 : A_i en double précision
- 1979 - 1980 : Racine carrée de A_i en double précision
- 1981 - 1982 : L'inverse de cette racine .
- 1983 - 1984 : Résultat de la vérification.

2.31.5 - ANALYSE DU PROGRAMME

- (1) 1600 70 1950 1650 Les contenus de la carte entrée sont chargés dans la zone 1951 - 1960.
La carte n'est pas chargement.
- 1650 69 1603 1606
- 1606 24 1732 1585 (2) (8001) = 00 0001 0000 + : distance entre Mot 1 et Mot 2 chargée en 1732.
- (2) 1585 65 1588 1593
- 1593 10 1596 1764 (8003 - 2) = 6515991653 0019510000 +
- | |
|--------|
| 1764.. |
| A.D.P. |
| 1740 |
- Transfert de A_i dans le PA
(1740) = (1596)
- (2.0) 1596 65 1599 1653
- 1653 10 1656 1760 (8003 - 2) = 6515991604 0019570000 +
- | |
|--------|
| 1760 |
| A.D.P. |
| 1740 |
- Division par $x_i \cdot P$
(1740) = (1656)
- 1656 65 1599 1604
- 1604 10 1607 1757 (8003 - 2) = 6516101565 0019570000 +
- | |
|--------|
| 1757 |
| A.D.P. |
| 1740 |
- Addition de $x_i \cdot P$
(1740) = (1607)
- 1607 65 1610 1565
- 1565 10 1568 1761 (8003 - 2) = 6515711575 0015500000 +

1761
A.D.P.
1740

1568 65 1571 1575

1575 10 1578 1763

1763
A.D.P.
8003

8003 65 1599 1654

1654 10 1657 1758

1758
A.D.P.
1740

1657 45 1660 1561

(3) (4)

(3) 1660 69 1979 1682

1682 24 1957 1612

1612 69 1980 1633

1633 24 1958 1585 (2)

(4) 1561 69 1979 1582

1582 24 1957 1611

1611 69 1980 1583

1583 24 1958 1661

1661 65 1564 1569

1569 10 1572 1764

1764
A.D.P.
1740

1572 65 1571 1625

1625 10 1628 1760

1760
A.D.P.
1740

1628 65 1581 1635

1635 10 1638 1763

Multiplication par +0,5 ; on obtient x_{p+1}
(1740) = (1568)

(8003 - 2) = 6515991654 0019790000 +

Transfert de x_{p+1} dans 1979-80

(8003 - 2) = 4516601561 001957 0000 +

Soustraction de x_i, p
(1740) = (1657)

1660 : $x_i, p+1 - x_i, p \neq 0$ Nouvelle itération

1561 : $x_i, p+1 - x_i, p = 0$ Fin des itérations

Transfert de $x_i, p+1$ (1979 - 80) à la place de x_i, p dans 1957 - 58
avant de reprendre une nouvelle itération

Transfert de $\sqrt{A_i}$ (1979 - 80) en 1957 - 58 pour jouer le rôle de
 $x_{i+1, 0}$ dans le calcul de $\sqrt{A_{i+1}}$

(8003 - 2) = 6515711625 00 1552 0000 +

Transfert de +1 dans le P.A.

(1740) = (1572)

(8003 - 2) = 65 1581 1635 00 1979 0000 +

Division par $\sqrt{A_i}$
(1740) = (1628)

(8003 - 2) = 65 1571 1675 00 1981 0000

1763
A.D.P.
8003

8003 65 1571 1675

1675 10 1678 1764

Transfert de $1 / \sqrt{A_i}$ en 1981 - 82

(8003 - 2) = 65 1571 1576 00 1979 0000

1764
A.D.P.
1740

1678 65 1571 1576

1576 10 1579 1761

Transfert de $\sqrt{A_i}$ dans le P.A
(1740) = (1678)

(8003 - 2) = 65 1632 1587 00 1979 0000

1761
A.D.P.
1740

1579 65 1632 1587

1587 10 1590 1763

Multiplication par $\sqrt{A_i}$
(1740) = (1579)

(8003 - 2) = 65 1588 1640 00 1983 0000

1763
A.D.P.
8003

8003 65 1588 1643

1643 10 1646 1764

Transfert de $(\sqrt{A_i})^2$ dans 1983 - 84

(8003 - 2) = 65 1649 1605 00 1951 0000

1764
A.D.P.
1740

1646 65 1649 1605

1605 10 1608 1763

Transfert de A_i dans le P.A
(1/40) = (1646)

(8003 - 2) = 65 1562 1567 00 1977 0000

1763
A.D.P.
8003

8003 65 1562 1567

1567 10 1570 1758

Transfert de A_i dans 1977 - 78 sur la zone de perfo.

(8003 - 2) = 45 1574 1626 00 1955 0000 +

1758
A.D.P.
1740

1570 45 1574 1626

(5) (6)

Squstraction de A_n
(1740) = (1570)

1574 : $A_i - A_n \neq 0$ poursuite de la table

1626 : $A_i - A_n = 0$ fin de la table

(5) 1574 65 1588 1693

1693 10 1696 1764

(8003 - 2) = 65 1699 1655 00 1951 0000

1764
A.D.P.
1740

Transfert de A_i dans le P.A
 $(1740) = (1696)$

1696 65 1699 1655

1655 10 1658 1757

$(8003 - 2) = 65 \ 1588 \ 1594 \ 00 \ 1953 \ 0000$

1757
A.D.P.
1740

Addition de h
 $(1740) = (1658)$

1658 65 1588 1594

1594 10 1597 1763

$(8003 - 2) = 71 \ 1977 \ 1596 \ 00 \ 1951 \ 0000$

1763
A.D.P.
8003

Transfert de $A_i + h = A_{i+1}$ en 1951 - 52

8003 71 1977 1596 (2.0)

Perforation du résultat.
 Renvoi pour calcul de $\sqrt{A_{i+1}}$ déjà chargé dans le P.A

(6) 1626 71 1977 1600 (1)

Perforation du résultat.
 Renvoi à la lecture d'une nouvelle carte.

2.32 - CALCUL DE e^{x+iy} EN DOUBLE PRECISION

Etant donné le nombre complexe $z = x + iy$, en double précision, calculer e^z en double précision également.

2.32.1 - METHODE UTILISEE

e^z est donné par le développement en série :

$$e^z = \sum_{p=0}^{\infty} \frac{z^p}{p!} \quad \text{série convergente à partir d'un certain rang.}$$

Le calcul se fera en 2 étapes :

1 - Calcul de la suite des u_p : On calcule $u_0 = 1$, $u_1 = z$, $u_2 = \frac{z^2}{2!}$ $u_p = \frac{z^p}{p!}$

en utilisant la formule de récurrence : $u_{p+1} = \frac{u_p}{p+1} \cdot z$

et jusqu'à ce qu'un premier terme u_n ait son module inférieur à un certain nombre ϵ choisi :

$$|u_n| < \epsilon \quad \text{ou} \quad |u_n|^2 < \epsilon^2$$

2 - Sommation de tous les u_p calculés précédemment et en commençant par le plus petit, donc u_n , de sorte que l'on obtient :

$$\sum_{p=n}^{\infty} u_p = \sum_{p=n}^{\infty} \frac{z^p}{p!} = e^z$$

2.32.2 - ORGANIGRAMME

(Voir page suivante)

2.32.3 - UTILISATION DU PROGRAMME

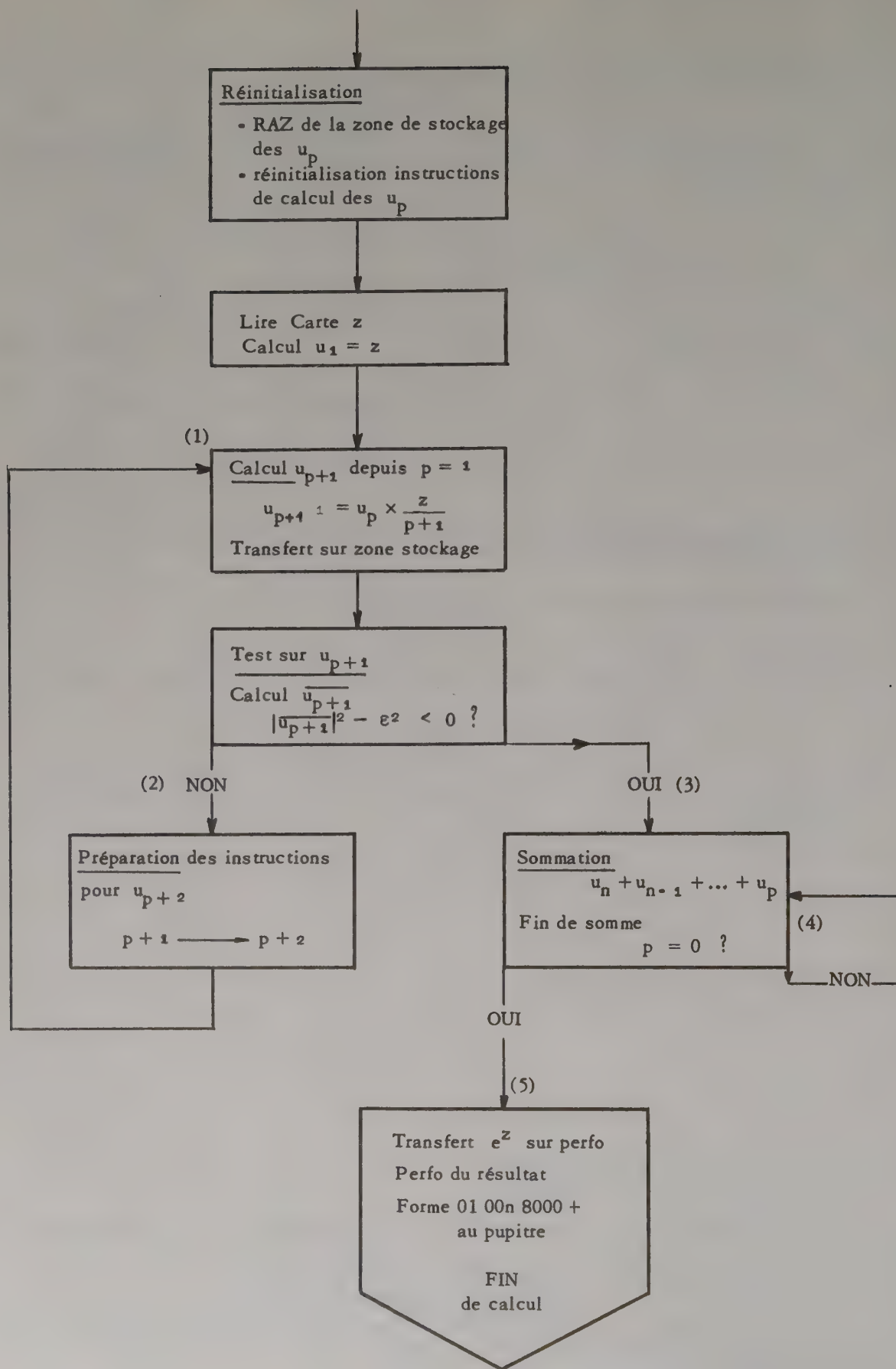
1 - Caractéristiques du programme de e^z

Il se présente sous la forme 1 Mot par carte (cartes jaunes) avec carte départ calcul automatique en 0800.

Il occupe les zones : 0750 - 0929 : programme de calcul
0930 - 0949 : constantes numériques

0950 - 1350 : 400 mémoires pour le stockage des 100 u_p

Il y a 77 mémoires de libres dans la zone 0750 - 0930.



2 • Format de la carte entrée

La zone d'entrée est prise dans la bande 1950 (1951 - 1960) , la carte est non chargement.

Mot 1 et 2 : partie réelle de z en double précision
 Mot 3 et 4 : partie imaginaire de z en double précision
 Mot 5 et 8 : 00 0000 0000 +

3 • Mode d'emploi

- Placer l'Arithmétique Complexe Double Précision (ACx DP) (cartes vertes à 6 Mots par carte) précédée de sa carte remise à zéro et du programme de chargement.
- Ensuite, placer le programme de calcul de e^z (cartes jaunes) et enfin la ou les cartes z pour lesquelles le calcul doit être fait.

Inscrire 70 000 4 xxxx + au pupitre pour la remise à zéro. Appuyer sur REST. GENE , DEPART PROG et DEPART LECTURE. Après la remise à zéro, inscrire 70 1951 xxxx + et appuyer sur DEPART PROG. Le calcul de la première carte z démarre automatiquement.

Après chaque calcul, c'est-à-dire après chaque carte z , le déroulement du programme est arrêté avec inscription au pupitre du nombre n de termes calculés sous la forme :

01 00n 0800

Pour répartir il suffit d'appuyer sur REST. GENE et DEPART PROG.

2.32.4 - RESULTATS

1 • Format des cartes sorties

La zone de sortie est celle de la bande 1950 (1977 - 1986).

Mot 1 et 2 : partie réelle de z en double précision
 Mot 3 et 4 : partie imaginaire de z en double précision
 Mot 5 et 6 : partie réelle de e^z en double précision
 Mot 7 et 8 : partie imaginaire de e^z en double précision.

Egalement, au pupitre : 01 00n 8000 + , n étant le nombre des termes calculés.

2 • Temps de calcul

Il dépend essentiellement du module de z et varie de 2 à 10 minutes environ.

3 • Précision sur le résultat

$$\text{On a } e^z = e^{x+iy} = e^x \cos y + i e^x \sin y$$

Il est donc facile de vérifier l'exactitude du résultat en calculant séparément $\cos y$, $\sin y$, e^x en triple précision et en formant les produits pour obtenir partie réelle et imaginaire de e^z .

Cette précision dépend essentiellement du nombre de termes calculés dans le développement de e^z . Si l'on désire 18 chiffres significatifs sur le résultat, il faut que le premier terme négligé soit inférieur à 10^{-18} . Donc $\varepsilon \leq 10^{-18}$ et $|u_p|^2 \leq \varepsilon^2 \leq 10^{-36}$

Le calcul fait avec $z = 1 + 1,2 i$ donne en double précision :

$$e^z = 5098 \dots 6040 + 5125 \dots 8929 i$$

et en triple $5098 \dots 603667 \dots + 5125 \dots 892914 \dots i$

pour un ε^2 pris égal à 5×10^{-36} .

Mais, étant donné la méthode de calcul utilisée, l'erreur croît très rapidement lorsque le module de z augmente :

pour $z = 1 + 8i$, ($|z| = 8,06$), l'erreur devient : sur la partie réelle 1500 unités du dernier ordre et 50 sur la partie imaginaire.

4 - Limites d'emploi

Puisque l'on ne dispose que d'une zone de stockage de 400 mémoires, correspondant à un développement de 100 termes, il faut que :

$$|u_{100}| \leq 10^{-18}$$

$$\text{d'où } \frac{|z^{100}|}{100!} \leq 10^{-18} \text{ donc } |z|^{100} \leq 9,333 \times 10^{139}$$

$$\text{donc } |z| \leq 10^{1,39} \times (9,333)^{\frac{1}{100}}$$

$$|z| \leq 25,10$$

Il faut donc que z reste à l'intérieur du cercle origine de rayon 25,10.

Remarquons que dans le développement de e^z , les u_p vont en croissant jusqu'à un certain rang avant de décroître et de tendre vers zéro. Dans le cas où $z = 25$, le maximum pour les u_p est atteint au voisinage de u_{25} et est de l'ordre de 5×10^9 , de sorte qu'il n'y a aucun dépassement de capacité à craindre dans le calcul des u_p .

Si l'on désire que $|z| > 25$, il faut disposer d'une zone de stockage plus vaste.

1 - Mémoires de travail à contenus constants

1.1 - Constantes numériques

$$\left. \begin{array}{l} 0950 : 51\ 1000\ 0000 + \\ 0951 : 00\ 0000\ 0000 + \\ 0952 : 00\ 0000\ 0000 + \\ 0953 : 00\ 0000\ 0000 + \end{array} \right\} \text{Nombre réel} + 1 \text{ en D.P.}$$

0938 : 15 5000 0000 +	}	Nombre réel ϵ^2 , en double précision $\epsilon^2 = 5 \times 10^{-36}$
0939 : 00 0000 0000 +		
0940 : 00 0000 0000 +		
0941 : 00 0000 0000 +		

0788 : 00 0002 0000 +	Distance entre partie réelle et imaginaire
0787 : 00 0001 0000 +	Distance entre 2 Mots d'une même partie

0772 : 00 0001 0000 +	
0799 : 51 2000 0000 +	Mot 1 du nombre réel + 2, premier diviseur des u_p
0826 : 00 0004 0000 +	Distance entre les Mots 1 de 2 u_p consécutifs.
0860 : 40 0000 0000 +	Diviseur

1.2 - Adresses facteurs constantes

0753 : 21 0954 0850 +	1 ^{re} instruction de remise à zéro de la zone de stockage des u_p
0803 : 21 1350 0850 +	dernière instruction de remise à zéro de la zone de stockage des u_p
0764 : 00 0954 0000 +	adresse de départ dans le calcul des u_2
0776 : 00 0958 0000 +	adresse de transfert de u_2
0801 : 00 0950 0000	adresse de u_0 , limite de sommation des u_p
0812 : 00 1981 0000	adresse du Mot 1 de e^z sur mémoire de perforation.
0831 : 00 1951 0000	adresse du Mot 1 de z
0841 : 00 0930 0000	adresse du Mot 1 du dénominateur $p + 1$
0804 : 00 0938 0000	adresse du Mot 1 de ϵ^2
0793 : 00 0934 0000	adresse du Mot 1 de $\overline{u_p}$

2 - Mémoires de travail à contenus variables

0934 • 35 • 36 • 37	conjugué de $u_p = \overline{u_p}$
0930 • 31 • 32 • 33	dénominateur $p + 1$
zone 0954 à 1349	zone de stockage de 100 u_p à 4 Mots
0770 : 00 (0950 + 4p) 0000	adresse de u_p pour le calcul de u_{p+1}
0782 : 00 (0954 + 4p) 0000	adresse de la zone de stockage des u_p où doit être chargé le u_{p+1} après son calcul.

2.32.5 • ANALYSE DU PROGRAMME

• Réinitialisation

0800 65 0753 8002	(8003 - 2) = 00 0000 0000 21 0954 + k 0850
0850 16 0803 0757	(0803) = 21 1350 0000
0757 45 0760 0761	
0760 15 8001 0769	
0769 15 0772 8002	(0772) = 00 0001 0000 (8002) = 21 0954 + k + 1 0850
0761 69 0764 0767	Remise à zéro de la mémoire suivante
0767 24 0770 0773	(8001) = 00 0954 0000 : démarrage du calcul des u_p
0773 69 0776 0779	
0779 24 0782 0785	(8001) = 00 0958 0000 : première adresse de transfert sur la zone de
0785 69 0788 0791	stockage
0791 24 1381 0784	(8001) = 00 0002 0000 : distance entre partie réelle et imaginaire
0784 69 0787 0790	
0790 24 1643 0796	(8001) = 00 0001 0000 : distance entre les 2 Mots de chaque partie
0796 65 0799 0853	
0853 20 0930 0783	(8003 - 2) = 00 0000 0000 51 2000 0000 +
0783 21 0931 0834	
0834 24 0932 0835	Transfert de + 2 , premier dénominateur en 0930 - 33
0835 24 0933 0786	

• Lecture et calcul de u_1

0786 70 1950 0900	
0900 69 1951 0754	
0954 24 1977 0780	
0780 24 0954 0807	
0807 69 1952 0755	
0755 24 1978 0781	Transfert de z sur mémoires de sortie 1977 - 80 et sur zone de stockage
0781 24 0955 0758	à la place de u_1 en 0954 - 57
0758 69 1953 0756	
0756 24 1979 0832	

0832 24 0956 0759

0759 69 1954 0857

0857 24 1980 0833

0833 24 0957 0810 (1)

- Calcul de u_{p+1} u_p est chargé en (0950 + 4p) et le diviseur (p + 1) en 0934 - 37

(1)0810 65 0770 0775

0775 10 0778 1773 (8003 - 2) = 65 0831 0885 00 (0950 + 4p) 0000 +

1773
ACxDP
1625

Transfert de u_p dans le PACx
(1625) = (0778)

0778 65 0831 0885

0885 10 0838 1771 (8003 - 2) = 65 0841 0795 00 1951 0000 +

1771
ACxDP
1625

Multiplication par z
(1625) = (0838)

0838 65 0841 0795

0795 10 0798 1770 (8003 - 2) = 65 0782 0837 00 0930 0000 +

1770
ACxDP
1481

Division par (p+1) ; on a donc $u_p \times \frac{z}{p+1} = u_{p+1}$ dans le PACx
(1481) = (0798)

0798 65 0782 0837

0837 10 0840 1772 (8003 - 2) = 69 1650 0903 00 (0954 + 4p) 0000 +

1772
ACxDP
8003

Transfert de u_{p+1} dans la zone de stockage à la suite de u_p en 0954 + 4p

- Test sur u_{p+1}

8003 69 1650 0903

0903 24 0934 0887 partie réelle de u_{p+1} transférée en 0934 - 0935

0887 69 1665 0768

0768 24 0935 0888

0888 61 1655 0809

0809 16 1670 0825

0825 21 0936 0789

0789 20 0937 0890

0890 65 0793 0797

0797 10 0751 1771

opposé de la partie imaginaire de u_{p+1} transférée en 0936 - 0937 de sorte que 0934 - 37 contient le conjugué de u_{p+1} , soit $\overline{u_{p+1}}$

(8003 - 2) = 65 0804 0859 00 0934 0000 +

1771
ACxDP
1625

Multiplication de u_{p+1} contenu dans le PACx par $\overline{u_{p+1}}$; on obtient le nombre réel $|u_{p+1}|^2$ (1625) = (0751)

0751 65 0804 0859

0859 10 0762 1769

(8003 - 2) = 60 1650 0805 00 0938 0000 +

1769
ACxDP
1625

Soustraction de ϵ^2 qui est réel. (PACx) = $|u_{p+1}|^2 - \epsilon^2$
(1625) = (0762)

0762 60 1650 0805

0805 15 1665 0819

0819 46 0822 0823

(8003 - 2) = $|u_{p+1}|^2 - \epsilon^2$

(2) (3)

0822 : négatif, donc u_{p+1} n'est pas le dernier terme de la suite

0823 : positif ou nul, donc u_{p+1} est le dernier terme de la suite.

• Préparation au calcul de u_{p+2}

(3) 0823 65 0782 0839

0839 24 0770 0873

0873 15 0826 0881

0881 20 0782 0836

0836 65 0841 0845

0845 10 0848 1773

(8001) = 00 (0954 + 4p) 0000 = 00 [0950 + 4 (p+1)] 0000

(0826) = 00 0004 0000

(8002) = 00 [0954 + 4 (p+1)] 0000 +

(8003 - 2) = 65 0801 0855 000 930 0000 +

1773
ACxDP
1625

Transfert de $(p+1)$ dans le PACx
(1625) = (0848)

0848 65 0801 0855

0855 10 0808 1768

$(8003 - 2) = 65\ 0841\ 0895\ 00\ 0950\ 0000$

1768
ACxDP
1625

Addition de +1 ; on obtient $(p+2)$
(1625) = (0808)

0808 65 0841 0895

0895 10 0810 1772

$(8003 - 2) = 65\ 0770\ 0775\ 00\ 0930\ 0000 +$

1772
ACxDP
8003

Transfert de $p+2$ en $0930 - 33$

(8003)

(1) Reprise des calculs pour u_{p+2}

• Sommation des u_p , le dernier calculé étant u_n

(2) 0822 65 0782 0889

0889 10 0792 1773

$(8003 - 2) = 65\ 0770\ 0875\ 00\ (0954 + 4n)\ 0000 +$

1773
ACxDP
1625

Transfert de u_n dans le PACx
(1625) = (0792)

0792 65 0770 0875

(2.0) 0875 10 0828 1768

$(8003 - 2) = 65\ 0770\ 0925\ 00\ (0950 + 4n)\ 0000 +$

1768
ACxDP
1625

Addition de u_{n+1}
(1625) = (0828)

0828 65 0770 0925

0925 16 0801 0905

$(0801) = 00\ 0950\ 0000$

0905 45 0858 0909

$(8002) = (0950 + 4k) - 950 = 4k$

(4) (5)

(0858) : le dernier terme additionné n'est pas u_0

0909 : le dernier terme additionné est u_0 fin des calculs.

(4) 0858 15 8001 0817

0817 16 0826 0882 (0826) = 00 0004 0000 +

0882 20 0770 0875 (2.0) (8002) = 00 (0950 + 4k - 4) 0000
= 00 [0950 + 4 (k-1)] 0000 +

• Fin des calculs

(5) 0909 65 0810 0867

0867 10 0820 1772 (8003 - 2) = 71 1977 0777 00 1981 0000 +

1772
ACxDP
8008

Transfert de $\sum_{p=0}^n u_p = e^z$ en 1981 - 84

8003 71 1977 0777

Perforation des résultats

0777 60 0782 0891

0891 11 0801 0806

(8003) = 00 4n 0000

0806 14 0860 0771

(0860) = 40 0000 0000

0771 30 0001 0827

(8002) = 00 00n 0000

0827 69 0830 0883

(0830) = 01 xxxx 8000

0883 22 0830 8001

8001 01 00n 0800

Fin des calculs.

CHAPITRE III

Règles de Programmation

SECTION 3.1

REGLES DE PROGRAMMATION

Les ~~macro-instructions~~ et les pseudo-instructions de compilation ont les mêmes structures que les instructions et pseudo-instructions PASO. Les cartes PASO restent donc utilisables pour perforer le programme initial, avant compilation.

3.11 - REGLES GENERALES

Lors de l'écriture en langage symbolique d'un programme destiné à être compilé, nous dirons un macro-programme, les règles de programmation PASO classiques restent valables, aux modifications suivantes près.

3.11.1 - CODE OPERATION

Tous les codes opération doivent être écrits sous forme symbolique, donc perforés dans les colonnes 48-49-50 de la carte PASO.

C'est sous cette forme symbolique que les nouveaux codes opération ont été créés, et c'est à cette condition que le Compileur analyse et interprète une instruction ou une macro-instruction. Autrement dit, une information dont le code opération est blanc ou de forme numérique (colonne 48 non perforée) est considérée comme une constante numérique par le Compileur et reproduite telle quelle.

3.11.2 - LISTE DES CODES-OPERATION INTERPRETES PAR LE COMPILEUR

Outre les codes opération standards, c'est-à-dire dont les équivalents en langage machine sont reconnus par l'I.B.M. 650 standard, le compileur permet d'employer les codes opérations des 6 tables suivantes.

• CODES C - Arithmétique Complexe Simple Précision

CAA : addition dans le PACx
CAS : soustraction dans le PACx
CDV : division du contenu du PACx
CMU : multiplication du contenu du PACx
CTR : transfert du contenu du PACx vers la mémoire générale
CZA : transfert du nombre dans le PACx
CZS : transfert de l'opposé du nombre dans le PACx
CZZ : chargement de la distance entre partie réelle et partie imaginaire des nombres complexes traités.

• CODES D - Arithmétique Réelle Double Précision

DAA : addition dans le PA
DAS : soustraction dans le PA
DAV : addition de valeur absolue dans le PA
DDV : division du contenu du PA
DMU : multiplication du contenu du PA
DSV : soustraction de valeur absolue dans le PA
DTR : transfert du contenu du PA vers la mémoire générale
DZA : transfert du nombre dans le PA
DZS : transfert de l'opposé du nombre dans le PA
DZV : transfert de la valeur absolue du nombre dans le PA
DZW : transfert de l'opposé de la valeur absolue du nombre dans le PA.
DZZ : chargement de la distance entre les 2 Mots des nombres réels traités.

• CODES G - Arithmétique Complexe Double Précision

GAA : addition dans le PACx
GAS : soustraction dans le PACx
GDV : division du contenu du PACx
GMU : multiplication du contenu du PACx
GTR : transfert du contenu du PACx vers la mémoire générale
GZA : transfert dans le PACx
GZS : transfert de l'opposé du nombre vers le PACx
GZZ : chargement de la distance entre parties réelle et imaginaire et de la distance entre les 2 Mots de chacune de ces parties des nombres complexes traités.

• CODES K - Fonctions en Double Précision

KFAT : Arc tangente	}	de l'argument double précision contenu dans 8003 - 8002
KFCN : Cosinus		
KFEX : Exponentielle		
KFLE : Logarithme népérien		
KFRC : Racine carrée		
KFSN : Sinus		

• CODES T • Arithmétique Réelle Triple Précision

TAA : addition dans le PA
TAS : soustraction dans le PA
TAV : addition de valeur absolue dans le PA
TDV : division du contenu du PA
TMU : multiplication du contenu du PA
TSV : soustraction de valeur absolue dans le PA
TTR : transfert du contenu du PA vers la mémoire générale
TZA : transfert du nombre dans le PA
TZS : transfert de l'opposé du nombre dans le PA
TZV : transfert de la valeur absolue du nombre dans le PA
TZW : transfert de l'opposé de la valeur absolue du nombre dans le PA
TZZ : chargement des distances entre Mot 1 et 2 et Mot 2 et 3 des nombres réels traités

• CODES V • Arithmétique Réelle Simple Précision

VAA : addition dans le PA
VAS : soustraction dans le PA
VAV : addition de valeur absolue dans le PA
VDV : division du contenu du PA
VMU : multiplication du contenu du PA
VSV : soustraction de valeur absolue dans le PA
VZA : transfert du nombre dans le PA
VZS : transfert de l'opposé du nombre dans le PA
VZV : transfert de la valeur absolue du nombre dans le PA
VZW : transfert de l'opposé de la valeur absolue du nombre dans le PA

3.11.3 • ADRESSES

L'adresse de l'instruction considérée peut être quelconque, blanche, symbolique ou numérique. L'adresse de la première instruction du bloc à compiler doit être non blanche.

L'adresse facteur

Sauf exception, elle doit être NUMERIQUE. Cette nécessité est due au fait que les nombres traités formés de plusieurs Mots doivent occuper des mémoires définies, d'adresse fixe ce qui ne serait pas possible en utilisant des adresses symboliques auxquelles PASO ferait correspondre un équivalent numérique, inconnu à priori, sans pour autant réserver l'emplacement des autres Mots du Nombre traité.

Exceptions

- 1 • Cas de l'Arithmétique réelle simple précision : les nombres traités étant constitués d'un seul Mot, l'adresse F peut être symbolique (mais non blanche).
- 2 • Cas des codes fonctionnels : l'adresse facteur de la macro instruction ne joue plus aucun rôle ; les colonnes 52-53-54-55 restent blanches alors que la colonne 51 reçoit le quatrième symbole alphabétique du code opération.

Adresse instruction à suivre

Elle peut être quelconque, blanche, symbolique ou numérique. La dernière instruction à suivre du bloc à compiler doit être non blanche.

3.11.4 - AUTRES CONSIDERATIONS

- Un type 8 ou 9, perforé en colonne 41 de la carte PASO, n'est pas interprété par le Compileur.
- Que le signe d'une macro-instruction soit plus (pas de perforation dans la colonne 42 de la carte PASO) ou moins (perforation arbitraire dans la colonne 42), ce signe n'est pas interprété et le Compileur fournit une séquence de traduction dont toutes les instructions sont positives. Ceci interdit tout artifice de programmation sur le signe d'une macro-instruction.
- Les remarques perforées dans les colonnes 63 à 72 de la carte PASO accompagnant une macro-instruction ne sont pas reproduites par le Compileur dans les cartes constituant la séquence de traduction de la macro-instruction.

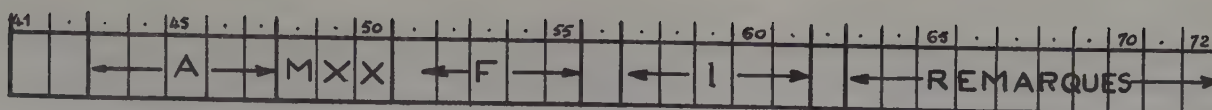
3.11.5 - REMARQUES

- 1 - L'artifice de programmation qui consiste à modifier, dans le déroulement d'un programme, le contenu d'une mémoire (A) pour l'utiliser dans de nouvelles conditions, est d'un emploi limité lorsqu'elle contient une macro-instruction. Différents cas sont d'ailleurs à considérer. On peut vouloir modifier dans la mémoire (A), soit l'adresse facteur (F), soit l'adresse instruction suivante (I), soit le code opération. Seule une modification portant sur l'adresse facteur (F) est possible (exception faite d'une macro-instruction fonctionnelle). Ces restrictions s'expliquent par le fait que l'on ne retrouve après compilation que l'adresse facteur (F) dans la mémoire (A), L'adresse instruction à suivre (I) se retrouve bien dans la séquence de traduction mais dans une mémoire dont, à priori, on ne connaît pas l'adresse.
- 2 - Il n'est pas permis de transformer en cours d'exécution d'un programme une instruction standard en une macro-instruction quelconque que le 650 ne saurait interpréter.
- 3 - Une macro-instruction de la liste initiale d'un programme est remplacée dans la liste compilée par une séquence de 3 instructions standards - (1 instruction pour les codes K). Cette liste compilée, destinée à l'assemblage par PASO, se trouve donc allongée par rapport à la liste initiale, dans la mesure où cette dernière renferme des macro-instructions. La longueur du programme est donc limitée d'autant.

3.12 - REGLES PARTICULIERES DE PROGRAMMATION

3.12.1 - CAS GENERAL DES MACRO - INSTRUCTIONS ARITHMETIQUES

La structure d'une macro-instruction est dans ce cas identique à celle d'une instruction standard PASO. Elle est de la forme :



• A est l'adresse de la mémoire du tambour contenant la macro-instruction ; A peut être blanc ou non, symbolique ou numérique.

• MXX est le macro-code arithmétique indiquant à la fois la nature de l'opération et la précision de l'Arithmétique utilisée. C'est le seul élément permettant l'identification d'une macro-instruction et il doit être l'un des 56 codes interprétables par le compilateur. MXX est formé de 3 symboles alphabétiques :

• M spécifie la précision et la nature réelle ou complexe des nombres traités • Si M =

- D : double précision réelle
- T : triple précision réelle
- V : simple précision réelle
- C : simple précision complexe
- G : double précision complexe

Les 2 autres symboles XX. précisent la nature de l'opération : AA : addition, DV: division, MU : multiplication etc ...

Le macro code MXX est nécessairement symbolique, sans équivalent numérique. Dans le cas contraire, la macro-instruction serait interprétée comme une constante numérique et reproduite telle quelle.

• F en position d'adresse facteur est l'adresse de la mémoire contenant le Mot 1 du nombre traité par la présente instruction F est nécessairement numérique, de la série 0000 - 1999 (en dehors de la zone de l'arithmétique). F ne peut pas être une adresse de la série 8000. La colonne 51 est donc blanche dans le cas général (voir exceptions dans le cas d'un code V ou K).

• L'adresse (I) de l'instruction à suivre est blanche ou non, numérique ou symbolique ; elle ne joue aucun rôle particulier.

• Les colonnes 41 (type 8 ou 9), 42 (signe) et 56 et 62 (index) sont quelconques mais non interprétées par le compilateur.

3.12.2 - EXCEPTIONS AU CAS GENERAL

1 - Cas des codes en V

Les nombres réels traités étant simple précision, ils n'occupent qu'une seule mémoire. Il est donc possible à F d'être symbolique.

2 - Cas des codes en ZZ

Ces macro-codes sont au nombre de 4 : CZZ , DZZ , GZZ , TZZ et les macro-instructions correspondantes sont :

(A)	CZZ	Δ	0000
(A)	DZZ	D'	0000
(A)	GZZ	Δ	D'
(A)	TZZ	D'	D''

Ces instructions indiquent que les quantités en position d'adresse facteur et d'adresse instruction sont les distances entre Mots des nombres traités. Ces codes sont utilisés comme tout autre code ; ils

doivent cependant être employés en début de programme avant tout calcul multiple précision. Si en cours de calcul, on est amené à modifier l'une ou l'autre de ces distances, il faut employer une nouvelle fois un code en ZZ.

L'adresse A est quelconque comme précédemment.

L'adresse facteur F est donc un nombre positif de 4 chiffres indiquant :

- la distance Δ entre partie réelle et imaginaire des nombres traités en arithmétique complexe : codes CZZ et GZZ.
- la distance D' entre Mot 1 et Mot 2 des nombres traités en arithmétiques réelles double ou triple précision : codes DZZ et TZZ.

La quantité en position d'adresse instruction suivante est également un nombre positif (ou nul) de 4 chiffres indiquant :

- la distance D' entre les 2 Mots de la partie réelle et de la partie imaginaire dans le cas de l'Arithmétique Complexe Double Précision : code GZZ.
- la distance D'' entre Mot 2 et Mot 3 dans le cas de l'Arithmétique Réelle Triple Précision : code TZZ.
- dans le cas des codes CZZ et DZZ, ces distances n'existant pas : on charge (0000) à la place de I.

Comme dans le cas de ces 4 macro-instructions l'adresse de l'instruction à suivre n'est pas indiquée, le Compileur enchainera automatiquement sur l'instruction qui suit celle-ci dans la liste initiale du macro-programme.

Les colonnes 51 et 67 doivent rester blanches.

3.12.3 - MACRO - INSTRUCTIONS FONCTIONNELLES : CODES K

Elles sont de la forme :



• A est l'adresse de la mémoire contenant la macro-instruction ; elle est blanche ou non, symbolique ou numérique.

• KFxx est le macro-code fonctionnel indiquant la nature de l'opération fonctionnelle à effectuer sur l'argument double précision initialement chargé en 8003 • 8002.

KF indique qu'il s'agit d'un macro-code fonctionnel double précision et xx spécifie la nature de la fonction. Exceptionnellement le macro-code est formé de 4 symboles alphabétiques et occupe les colonnes 48 • 49 • 50 et aussi 51 en position d'adresse facteur.

• La position de l'adresse facteur est blanche, sauf la colonne 51 qui reçoit le quatrième symbole du macro-code fonctionnel.

• I est l'adresse de l'instruction à suivre ; elle est blanche ou non, symbolique ou numérique.

• Les colonnes 41 (type 8 ou 9) , 42 (signe) , 56 et 62 (index) sont quelconques mais non interprétées par le compileur.

Remarque : Avant l'utilisation d'un macro-code K, l'argument double précision doit être chargé en 8003-8002. Après utilisation le résultat apparait dans 8003 - 8002 et le PA de l'Arithmétique Double Précision.

Aucune autre précaution particulière n'est à prendre.

3.12.4 - PSEUDO - INSTRUCTIONS

1 - Pseudo-instructions d'assemblage PASO

Elles peuvent figurer dans la liste initiale du programme; le compilateur les restitue telles quelles.

2 - Pseudo-instruction BBL

La liste initiale d'un programme ne peut être compilée sans qu'elle soit précédée d'une carte pseudo-instruction BBL .

Carte BBL de première espèce

Une pseudo-instruction BBL de première espèce est de la forme :

		48 49 50 51				
		BBL	A J			

Seules les positions 48 à 51 sont perforées ; les trois premières reçoivent le pseudo-code BBL et la colonne 51, l'une des 10 lettres A , B , C , D , E , F , G , H , I et J suivant la bibliothèque utilisée.

L'interprétation d'une pseudo-instruction BBL de première espèce a pour effet de préparer la réservation de la zone correspondant à la bibliothèque utilisée et la vérification en cours de compilage de la compatibilité des codes interprétés.

Une seule carte BBL de première espèce doit être utilisée.

Carte BBL de seconde espèce

Une pseudo-instruction BBL de seconde espèce est de la forme :

		48 49 50 51		57 58 59 60 61		
		BBL	X	K F x x		

Les positions 48 à 50 reçoivent le pseudo-code BBL . La position 51 reçoit l'une des quatre lettres A , B E et F suivant l'Arithmétique utilisée et les positions 57 à 60 l'un des six codes fonctionnels définis au § 3.11.2.

L'interprétation d'une pseudo-instruction BBL de seconde espèce a pour effet de préparer la réservation des zones correspondant à l'Arithmétique, à la fonction indiquée dans les positions 57 à 60 et la vérification en cours de compilage de la comptabilité des codes interprétés.

On peut utiliser une ou plusieurs (6 au maximum) cartes BBL de seconde espèce correspondant à une fonction différente ; cependant pour chacune de ces cartes les colonnes 48 à 51 doivent recevoir les mêmes perforations.

3 - Pseudo-instruction Z

Toute liste initiale d'un programme à compiler doit se clôturer par la pseudo-instruction Z dont la structure est la suivante :



Z est perforé dans la colonne 48 de la carte PASO.

L'interprétation de cette pseudo-instruction a pour effet de perforer une carte PDT indiquant en remarque le nombre de constantes internes symboliques qu'il a été nécessaire de construire en cours de compilation pour assurer l'enchaînement des séquences de traduction interprétable par PASO III.

Cette interprétation provoque également la réinitialisation du compilateur.

3.12.5 - REMARQUES GENERALES

- 1 - La première des adresses A et la dernière des adresses I de la liste à compiler doivent être non blanches.
- 2 - On peut inclure les constantes numériques du programme au bloc à compiler ; elles seront restituées telles quelles.

SECTION 3.2

BIBLIOTHEQUES

3.21 - BIBLIOTHEQUES MULTIPLE PRECISION

3.21.1 - NOMENCLATURE DES BIBLIOTHEQUES

A	: Arithmétique Réelle Double Précision (A.D.P.)	BBL A
B	: Arithmétique Réelle Mixte, Simple et Double Précision (A.SP.P.)	BBL B
C	: Arithmétique Réelle Triple Précision (A.T.P.)	BBL C
D	: Arithmétique Complexe Simple Précision (ACxSP)	BBL D
E	: Arithmétique Complexe Double Précision (ACxDP)	BBL E
F	: Arithmétique Complexe Mixte, Simple et Double Précision (A.Cx.SD.P.)	BBL F
G	: Bibliothèque A et fonctions double précision	BBL G
H	: Bibliothèque B et fonctions double précision	BBL H
I	: Bibliothèque E et fonctions double précision	BBL I
J	: Bibliothèque F et fonctions double précision	BBL J

Suivant le problème à résoudre et la nature des nombres à traiter on choisira donc l'une de ces 10 bibliothèques. (Voir paragraphe suivant),

3.21.2 - PLAN D'ENCOMBREMENT DES BIBLIOTHEQUES

BBL A	: 1700 - 1999	BBL F	: 1400 - 1999
BBL B	: 1650 - 1999	BBL G	: 0050 - 0845 et 1700 - 1999
BBL C	: 1550 - 1999	BBL H	: 0050 - 0845 et 1650 - 1999
BBL D	: 1650 - 1999	BBL I	: 0050 - 0845 et 1540 - 1999
BBL E	: 1540 - 1999	BBL J	: 0050 - 0845 et 1400 - 1999

Pour chacune de ces bibliothèques les bornes de zone d'encombrement sont incluses ; ainsi pour la BBL A, la première mémoire libre est 1699.

Dans chacune de ces zones, les zones d'entrée et sortie de la bande 1950 : 1950 - 1960 et 1977 - 1986 sont gardées libres.

Fonctions

Arc tangente	:	KFAT	:	0645 - 0844
Cosinus	:	KFCN	:	0480 - 0644
Exponentielle	:	KFEX	:	0310 - 0479
Logarithme népérien	:	KFLE	:	0150 - 0309
Racine carrée	:	KFRC	:	0050 - 0149
Sinus	:	KFSN	:	0480 - 0644

3.22 - CHOIX DE LA BIBLIOTHEQUE

Le choix de la bibliothèque est essentiellement conditionné par la nature des calculs et des nombres à traiter.

3.22.1 - 1er CAS - CALCULS PUREMENT ARITHMETIQUES

Les seules opérations effectuées étant des opérations purement arithmétiques (ou des transferts), suivant la nature des nombres à traiter on fera choix de l'une des 6 bibliothèques suivantes :

Nombres réels, double précision	BBL A
Nombres réels, simple et double précision	BBL B
Nombres réels triple précision	BBL C
Nombres complexes simple précision	BBL D
Nombres complexes double précision	BBL E
Nombres complexes simple et double précision	BBL F

La première carte de la liste à compiler doit être une carte BBL de première espèce ; elle doit être unique.

BBL X

avec $X = A, B, C, D, E$ ou F

3.22.2 - 2ème CAS - UNIQUEMENT DES CALCULS DE FONCTIONS

Les calculs font appel aux fonctions double précision à l'exclusion de tout autre opération arithmétique ou transfert. On utilise la bibliothèque G, et la première carte de la liste à compiler est :

BBL G

de première espèce et unique.

3.22.3 - 3ème CAS - CALCULS DE FONCTIONS ET CALCULS ARITHMETIQUES

Remarquons tout de suite que les seuls calculs de fonctions que l'on puisse effectuer sont des calculs en double précision. Suivant la nature des nombres traités arithmétiquement, on utilisera :

Nombres réels double précision et fonctions double précision	BBL G
Nombres réels simple et double précision et fonctions double précision	BBL H
Nombres complexes double précision et fonctions double précision	BBL I
Nombres complexes simple et double précision et fonctions double précision	BBL J

La première carte de la liste à compiler doit être une carte BBL de première espèce et elle doit être unique:

BBL X

avec X = G, H, I ou J

3.22.4 - 4ème CAS - CALCULS PRINCIPALEMENT ARITHMETIQUES - UTILISATION

D'UN NOMBRE RESTREINT DE FONCTIONS DOUBLE PRECISION

On utilise d'une part, la bibliothèque fonctionnelle, dont seules les zones correspondant aux fonctions utilisées seront réservées, et d'autre part, l'une des quatre bibliothèques suivant la nature des nombres traités.

Nombres réels double précision	BBL A
Nombres réels simple et double précision	BBL B
Nombres complexes double précision	BBL E
Nombres complexes simple et double précision	BBL F

La ou les premières cartes de la liste à compiler devront être des cartes BBL de seconde espèce, il y aura autant de cartes que de fonctions différentes utilisées, le code de la fonction étant perforé dans les colonnes 57 à 60 de la carte. L'indication de bibliothèque (A, B, E ou F) doit rester la même pour chacune de ces cartes.

Voir exemple au § 5.43 : utilisant des nombres réels double précision et les seules fonctions cosinus, exponentielle et sinus, la liste à compiler doit avoir les 3 cartes suivantes en tête :

BBL A	KFCN
BBL A	KFEX
BBL A	KFSN

3.22.5 - REMARQUES

- 1 - La dernière carte de la liste à compiler doit toujours être une carte Z . Si cette liste comporte des constantes, la carte Z doit être placée après ces constantes.
- 2 - La première adresse (A) et la dernière adresse (I) de la liste à compiler doivent être non blanches.
- 3 - A chaque fois que l'on utilise une arithmétique pour nombres complexes, il est également possible de l'utiliser pour le traitement de nombres réels de même précision .

3.23 • SOUS • PROGRAMMES ENTRANT DANS LA COMPOSITION DES BIBLIOTHEQUES

Ce sont :

Les Arithmétiques Réelles Double et Triple Précision,
Les Arithmétiques Complexes Simple et Double Précision,
L'Arithmétique Mixte Réelle et l'Arithmétique Mixte Complexe,
et enfin Les sous-programmes fonctionnels : Arctangente, cosinus, exponentielle, logarithme népérien, racine carrée et sinus.

Nous n'entrerons pas à nouveau dans les détails des sous-programmes Arithmétiques qui ont déjà été étudiés; nous nous bornerons à donner quelques explications en vue d'une macro-programmation par utilisation de macro-codes opérations relativement à chacun de ces sous-programmes arithmétiques.

3.23.1 - ARITHMETIQUES REELLES

1 - Arithmétique Réelle Double Précision

1.1 - Nombres traités

Les nombres réels traités sous forme Virgule Flottante Double Précision, à exposant de 2 chiffres et mantisse de 18 sont distribués en 2 Mots 650 chargés dans 2 mémoires d'adresses F et F' telles que $F' - F = D'$, D' étant une distance positive, généralement fixée pour tout un programme de calcul.

1.2 - Programmation pour Compilateur

1) - Réservation : L'Arithmétique Réelle Double Précision constituant la bibliothèque A, la première carte du bloc à compiler devra être BBL A qui réservera automatiquement la zone de l'Arithmétique (1700 - 1999) .

2) - Chargement de la distance D' : Avant toute opération Double Précision, il faut utiliser une macro-instruction.

(A) DZZ (D') 0000
qui chargera D' en 1732.

3) • Instructions de calcul : Elles sont de la forme : (A) Dxx (F) (I) , F étant nécessairement numérique de la série 0000-1699.

Si Dxx est un code de transfert vers le PA (DZA , DZS , DZV ou DZW) , les contenus de F et F' sont chargés dans le PA.

Si Dxx est un code arithmétique, (DAA , DAS , DAV , DDV , DMU ou DSV) l'opération indiquée par ce code opération se fait entre, d'une part le premier opérande précédemment chargé dans le PA et d'autre part le second opérande chargé dans les mémoires F et F' ; F étant l'adresse indiquée en position d'adresse facteur de la macro-instruction.

Dans ces 2 cas, après chaque opération, les contenus du PA et du PD sont modifiés et le résultat apparaît en 8003 - 2 .

Si Dxx = DTR, le contenu du PA est transféré dans les mémoires F et F'.

L'instruction à suivre est prise en I .

2 • Arithmétique Réelle Mixte, Simple et Double Précision

2.1 • Nombres traités

Les nombres réels traités en Virgule Flottante Simple Précision à exposant de 2 chiffres et mantisse de 8 , constituent un Mot 650 chargé dans une seule mémoire d'adresse F.

Les nombres réels traités en Virgule Flottante Double Précision à exposant de 2 chiffres et mantisse de 18 chiffres sont distribués en 2 Mots 650, chargés dans 2 Mémoires d'adresses F et F' telles que $F' - F = D$, D' étant une distance positive, généralement fixée pour tout un programme de calcul.

2.2 • Programmation en Simple Précision pour Compilateur

1) • Réservation : L'Arithmétique Réelle Mixte constituant la bibliothèque B, la première carte du bloc à compiler devra être BBL B qui réservera automatiquement la zone de l'Arithmétique (1650 - 1999).

2) • Instruction de calcul : Elles sont de la forme : (A) Vxx (F) (I) où F peut être numérique ou symbolique.

Si Vxx est un code de transfert vers le PA (VZA , VZS , VZV , VZW) le contenu de F est chargé dans le PA.

Si Vxx est un code arithmétique (VAA , VAS , VAV , VDV , VMU , VSV) , l'opération indiquée par ce code opération se fait entre, d'une part le premier opérande précédemment chargé dans le PA et d'autre part, le second opérande chargé dans la mémoire F , indiquée en position d'adresse facteur de la macro-instruction.

Dans ces 2 cas, après chaque opération, les contenus du PA et PD sont modifiés et le résultat apparaît en 8003. Rappelons que 8003 ne joue plus le rôle d'Accumulateur pour l'Arithmétique mais que ce sont les mémoires 1700-1715-1730 qui jouent ce rôle.

Le transfert du contenu du PA dans la mémoire F se fait en utilisant un code : TRG

L'instruction suivante est prise en I.

2.3 - Programmation en Double Précision pour Compilateur

1) - Réservation : Utiliser comme première carte BBL B.

2) - Chargement de la distance D' : Avant toute opération Double-Précision, il faut utiliser une macro-instruction.

(A) DZZ (D') 0000

qui chargera D' en 1732.

3) - Instructions de calcul : Elles sont de la forme : (A) Dxx (F) (I) où F est nécessairement numérique de la série 0000 - 1649 - (Cf Double Précision).

L'instruction suivante est prise en I.

3 - Arithmétique Réelle Triple Précision

3.1 - Nombres traités

Les nombres réels traités sous forme Virgule Flottante Triple Précision, à exposant de 3 chiffres et mantisse de 27, sont distribués en 3 Mots 650 chargés dans 3 mémoires d'adresses F, F' et F'' telles que :

$$F' - F = D' \quad \text{et} \quad F'' - F' = D''$$

D' et D'' étant des distances positives, généralement fixées pour tout un programme de calcul.

3.2 - Programmation pour Compilateur

1) - Réservation : L'Arithmétique Réelle Triple Précision constituant la bibliothèque C, la première carte devra être BBL C qui réservera automatiquement la zone de l'Arithmétique (1550 - 1999).

2) - Chargement des distances D' et D'' : Avant toute opération Triple Précision, il faut utiliser une macro-instruction :

(A) TZZ (D') (D'')

qui chargera D' en 1732 et D'' en 1747.

3) - Instructions de calcul : Elles sont de la forme : (A) Txx (F) (I), F étant nécessairement numérique de la série 0000 - 1549.

Si Txx est un code de transfert vers le PA (TZA, TZS, TZV, TZW) les contenus de F, F' et F'' sont chargés dans le PA.

Si Txx est un code arithmétique (TAA, TAS, TAV, TDV, TMU ou TSV) l'opération indiquée par le code opération se fait entre, d'une part le premier opérande précédemment chargé dans le PA et d'autre part le second opérande chargé dans les mémoires F, F' et F'' ; F étant l'adresse indiquée en position d'adresse facteur de la macro-instruction.

Dans ces 2 cas, après chaque opération, les contenus du PA et du PD sont modifiés et le résultat de l'opération apparaît en 8003-2-1.

Si Txx = TTR, le contenu du PA est transféré dans les mémoires F, F' et F''.

L'instruction suivante est prise en I.

3.23.2 • ARITHMETIQUES COMPLEXES

1 • Arithmétique Complexe Simple Précision

1.1 • Nombres traités

Les nombres complexes traités sous forme Virgule Flottante Simple Précision, à partie réelle et partie imaginaire composées d'un exposant de 2 chiffres et d'une mantisse de 8, sont distribuées en 2 Mots 650, un par partie, chargés dans 2 mémoires d'adresses F et G telles que :

$$G - F = \Delta$$

Δ étant la distance positive entre partie réelle et partie imaginaire et généralement fixée pour tout un programme de calcul.

1.2 • Programmation pour Compilateur

1)- Réservation : L'Arithmétique Complexe Simple Précision constituant la bibliothèque D, la première carte du bloc à compiler devra être BBL D qui réservera automatiquement la zone de l'Arithmétique (1650 - 1999).

2) • Chargement de la distance Δ : Avant toute opération, il faut utiliser une macro-instruction (A) CZZ (Δ) 0000 qui chargera Δ en 1732.

3) • Instructions de calcul : Elles sont de la forme (A) Cxx (F) (I) , F étant nécessairement numérique de la série 0000 - 1649.

Si Cxx est un code de transfert vers le PACx (CZA ou CZS) les contenus de F et F' sont chargés dans le PACx.

Si Cxx est un code arithmétique (CAA , CAS , CDV ou CMU) l'opération indiquée par ce code opération se fait entre, d'une part, le premier opérande précédemment chargé dans le PACx et d'autre part, le second opérande chargé dans les mémoires F et G , F étant l'adresse indiquée en position d'adresse facteur de la macro-instruction.

Dans ces 2 cas, après chaque opération, les contenus du PACx et du PDCx sont modifiés.

Si Cxx = CTR, le contenu du PACx est transféré dans les mémoires F et G.

L'instruction à suivre est prise en I.

2 • Arithmétique Complexe Double précision

2.1 • Nombres traités

Les nombres complexes traités sous forme Virgule Flottante Double Précision, à partie réelle et partie imaginaire composées d'un exposant de 2 chiffres et d'une mantisse de 18 chiffres distribuées chacune en 2 Mots 650, sont chargés dans 4 mémoires d'adresses F et F', G et G' telles que :

$$\begin{aligned} G - F &= G' - F' = \Delta \\ F' - F &= G' - G = D' \end{aligned}$$

Δ étant la distance positive entre partie réelle et partie imaginaire, D' la distance entre les 2 Mots de chacune de ces parties et généralement fixes pour tout un programme de calcul.

2.2 - Programmation pour Compileur

1) - Réservations : L'Arithmétique Complexe Double Précision constituant la bibliothèque E, la première carte du bloc à compiler devra être BBL. E qui réservera automatiquement la zone de l'Arithmétique. (1540 - 1999)

2) - Chargement des distances Δ et D' : Avant toute opération, il faut utiliser la macro-instruction.

(A) GZZ (Δ) (D')

qui chargera Δ en 1541 et D' en 1581.

2.3 - Instructions de calcul

Elles sont de la forme : (A) Gxx (F) (I) , F étant nécessairement numérique de la série 0000 - 1539.

Si Gxx est un code de transferts vers le PACx (GZA ou GZS), les contenus de F, F' G et G' sont chargés dans le PACx.

Si Gxx est un code arithmétique (GAA, GAS, GDV, G.MU) l'opération indiquée par ce macro-code se fait entre, d'une part le premier opérande précédemment chargé dans le PACx et d'autre part, le second opérande chargé dans les mémoires F, F', G et G', F étant l'adresse indiquée en position d'adresse facteur de la macro-instruction.

Dans ces 2 cas, après chaque opération, les contenus du PACx et du PDCx sont modifiés.

Si Gxx = GTR, le contenu du PACx est transféré dans les mémoires F, F', G et G'.

L'instruction suivante est prise en I.

3 - Arithmétique Complexe Mixte Simple et Double Précision

3.1 - Nombres traités

Les nombres complexes traités en Virgule Flottante Simple Précision, à partie réelle et imaginaire formées d'un exposant de 2 chiffres et d'une mantisse de 8, sont distribués en 2 Mots 650, un pour chacune des parties, chargés dans 2 mémoires d'adresses F et G telles que :

$$G - F = \Delta$$

Les nombres complexes traités en Virgule Flottante Double précision, à partie réelle et imaginaire formées d'un exposant de 2 chiffres et d'une mantisse de 18 chiffres, sont distribués en 2 fois 2 Mots 650, deux pour chacune des parties, chargés dans 4 mémoires d'adresses F, F' et G, G' telles que

$$\begin{aligned} G - F &= G' - F' = \Delta \\ F' - F &= G' - G = D' \end{aligned}$$

Δ étant la distance positive entre partie réelle et partie imaginaire et D' la distance entre le Mot 1 et le Mot 2 (s'il existe) de chacune de ces 2 parties. Ces distances sont généralement conservées fixes tout au long d'un programme de calcul.

3.2 - Programmation en Simple Précision pour Compilateur

1) - Réservation : L'Arithmétique Complexe Mixte constituant la bibliothèque F, la première carte du bloc à compiler devra être BBLF qui réservera automatiquement la zone de l'Arithmétique (1400 - 1999).

2) - Chargement de Δ : Avant toute opération, il faut utiliser la macro-instruction

(A) CZZ Δ 0000

qui charge Δ en 1431 et 00 0000 0000 + en 1441.

3) - Instructions de calcul : Voir Arithmétique Complexe Simple Précision, l'adresse F étant nécessairement numérique de la série 0000 - 1399.

3.3 - Programmation en Double Précision pour Compilateur

1) - Réservation : La première carte du bloc à compiler doit être BBL F.

2) - Chargement de Δ et D' : Avant toute opération, il faut utiliser la macro-instruction

(A) GZZ Δ D'

qui charge Δ en 1431 et D' en 1441.

3) - Instructions de calcul : Voir Arithmétique Complexe Double Précision, l'adresse F étant nécessairement numérique de la série 0000 - 1399.

3.23.3 - BIBLIOTHEQUE FONCTIONNELLE

1 - Argument

L'argument est un nombre réel, exprimé en double précision Virgule Flottante Flair, formé d'un exposant de 2 chiffres, d'une mantisse de 18 et distribué en 2 Mots 650 chargés dans 2 mémoires d'adresses F et F' telles que :

$$F' - F = D'$$

où D' est la distance entre les Mots 1 et 2 et reste généralement fixe tout au long du programme de calcul.

2 - Programmation

2.1 - Réserve

Si l'on fait appel à l'ensemble de la bibliothèque fonctionnelle , on utilisera l'une des quatre bibliothèques :

G , H , I ou J

La première carte de la liste à compiler devra être une carte BBL de première espèce:

B B L G (ou H , I , J)

qui se traduira par la réserve des zones correspondant à l'Arithmétique et à la totalité de la bibliothèque fonctionnelle .

Si l'on ne fait appel qu'à certaines fonctions , on utilisera l'une des quatre bibliothèques

A , B , E ou F

suivant la nature des nombres à traiter , et à la partie utile de la bibliothèque fonctionnelle.

La première carte de la liste à compiler devra être une carte BBL de seconde espèce, par exemple :

B B L A K F A T

Il y aura autant de cartes de cette forme que l'on utilise de fonctions différentes . Pour toutes ces cartes , l'indication de bibliothèque devra rester la même . La traduction de ces cartes réservera les zones correspondant à l' Arithmétique et à chacune des fonctions indiquées .

2.2 - Chargement de D'

Avant toute opération , il faut utiliser une macro-instruction :

(A) D Z Z D' O O O O

qui chargera D' en 1732 .

2.3 - Instructions de calcul

Avant d'utiliser une macro-instruction fonctionnelle , il est nécessaire que l'argument soit chargé en 8003 - 8002 . Pour ce faire , on peut utiliser la séquence :

$$\begin{cases} R A G & F \\ A A D & F' \end{cases}$$

ou si l'on programme en macro-instruction , D Z A F , qui chargera l'argument dans le P A mais aussi dans 8003 - 8002 , ce qui est essentiel .

La macro-instruction fonctionnelle s'écrit :

(A) K F x x (I)

K F x x étant l'un des six codes définis au paragraphe 3. 112 .

Après calcul , la valeur de la fonction pour l'argument considéré , se trouve chargée en 8003 - 2 , et également dans le P A de l' A.D.P .

Remarquons que le fait d'utiliser un macro-code fonctionnel ne modifie en rien le contenu initial de 1732 .

SECTION 3.3

GUIDE OPERATEUR

Le Compileur est assemblé sur le tambour de la mémoire 1000 à la mémoire 1999 . Il laisse libre dans cette zone 170 mémoires (voir le plan d'encombrement du compileur) .

Il est perforé en 168 cartes 6Mots , avec l'indicatif 06 3844 onnn en Mot 8 (cartes oranges). Ce paquet de cartes est précédé d'une carte de remise à zéro des mémoires du tambour et d'un programme de chargement 6 Mots spécial avec vérification de la numérotation des cartes . Il est suivi d'une carte de départ automatique de compilation .

Les 10 bibliothèques sont perforées en carte 6 Mots avec l'indicatif 109 185 onnn en Mot 8 (cartes vertes) . Chaque paquet de cartes est précédé d'un programme de chargement 6 Mots spécial avec vérification de la numérotation des cartes .

3.31 - 1er PASSAGE MACHINE : COMPILATION

3.31. 1 - PRATIQUE DE LA COMPILATION

1 - Préparation de la liste à compiler

Placer devant celle-ci une seule carte portant une macro-instruction de première espèce BBL A ou BBL J suivant la bibliothèque considérée , ou bien , 1 à 6 cartes portant la macro-instruction de seconde espèce de la forme BBL X KFxx avec X = A , B , E ou F .

Placer obligatoirement derrière celle-ci une carte portant la macro-instruction Z .

2 - Unité 533

Mettre en place le tableau de connexions PASO

Mettre des cartes vierges dans le magasin de perforation

Si le compileur n'est pas enregistré sur le tambour , mettre dans l'ordre , la carte remise à zéro , le programme de chargement 6 Mots , le compileur et le programme à compiler , dans le Magasin de

lecture .

3 - Pupitre 650

Positionnement des commutateurs :

- STOP sur « effectif »
- PROGRAMME sur « normal »
- CONTRÔLE sur « marche »
- Indicateur sur « Registre Programme »
- Dépassement de capacité sur « détection »
- ERREUR sur « arrêt »
- SIGNE +

4 - Mise en marche

Si le compilateur n'est pas déjà chargé sur le tambour :

- afficher 70 0004 xxxx au pupitre
- appuyer sur les boutons REST GENE et DEP. PROG .

La remise à zéro s'effectue . Pendant ce temps , on modifie le pupitre en affichant

70 1951 xxxx +

Dès la fin de la remise à zéro ,

- appuyer sur DEP. PROG. qui commande le chargement du compilateur , après quoi , la compilation débute automatiquement .

Si le compilateur est déjà chargé et que le pupitre affiche 70 1963 1961 , appuyer sur DEP. LECTURE.

Si le pupitre est quelconque , introduire 1300 dans le registre d'adresse et appuyer sur DEP. PROG. qui commande le départ de la compilation .

5 - Exécution de la compilation

Si la dernière carte lue par le compilateur est une carte Z , la compilation s'arrête sur l'instruction 70 1963 1961 . La réinitialisation ayant donc eu lieu , il suffit d'appuyer sur DEP. LECT. pour une nouvelle compilation .

Si la dernière carte lue n'est pas une carte Z , le compilateur s'arrête sur l'instruction 70 1963 1962 .

Plusieurs arrêts programmés signalent à l'opérateur plusieurs fautes de programmation .

Ce sont :

01 0301 1300 : La première carte lue n'est pas une carte BBL. Remplacer cette première carte et

appuyer sur DEP . PROG.

01 0302 1300 : La carte BBL lue est inconnue du compilateur . Modifier la carte BBL et appuyer sur DEP. PROG.

01 0303 1334 : La seconde carte lue est une carte BBL de première espèce . Supprimer cette carte et poursuivre la compilation en appuyant sur DEP. PROG .

01 0340 8000 : Il y a incompatibilité entre le macro - code analysé et la bibliothèque indiquée en 1ère carte . Il faut reconsidérer le programme et reprendre la totalité du compilage en 1300 .

01 0310 1334 : Il y a une erreur dans l'enchaînement des instructions ; par exemple : 2 adresses en blanc consécutives . Corriger l'instruction erronée , recharger cette carte corrigée dans le magasin et appuyer sur DEP . PROG .

01 0320 1493 : L'adresse A de la carte suivante n'est pas blanche , alors que l'adresse I de la carte présente l'est . Dans la liste initiale , reprendre les 2 instructions dont l'enchaînement est défectueux , les corriger . Dans la liste perforée , enlever les deux dernières cartes . Pour la poursuite du compilage, repartir avec les 2 cartes corrigées en tête et appuyer sur DEP . PROG .

01 0330 1751 : L'adresse facteur F est blanche ou bien symbolique , alors que le macro - code n'est pas un code V . Corriger la carte et reprendre la compilation à partir de la carte corrigée en appuyant sur DEP. PROG .

01 0311 8000 : 400 constantes internes symboliques ont été nécessaires à l'enchaînement des séquences de traduction . Lors de l'assemblage par PASO , la table des symboles sera nécessairement saturée ; il faut donc reprendre le programme , ou bien le diviser en 2 parties que l'on compilera et assemblera séparément .

01 0322 8000 : Dans le cas de l'utilisation d'une pseudo - instruction BBL de seconde espèce , il y a incompatibilité entre l'arithmétique perforée en colonnes 48 - 51 et la fonction indiquée en colonnes 57 -60. Corriger la carte et reprendre la totalité du compilage .

01 0344 8000 : Les deux cartes BBL de seconde espèce consécutives n'ont pas les mêmes perforations en colonnes 48 - 51 . Corriger et reprendre le compilage .

6 - Fin de Compilation

Si l'on a eu soin de placer une carte Z à la fin de la liste à compiler , l'arrêt se fait sur l'instruction 70 1963 1961 après réinitialisation du compilateur et perforation d'une dernière carte PDT indiquant en remarques le nombre de constantes symboliques internes utilisées pour le compilage de la liste initiale .

3.31.2 - RESULTATS DE LA COMPILATION

1 - Généralités

Dans la nouvelle liste perforée par le compilateur, on retrouve identiques à elles-mêmes, les cartes disponibilités, les cartes commentaires, pseudo-instruction PASO ; sont également reproduites les instructions PASO et les constantes numériques de la liste initiale.

Quant aux macro-instructions, leur traduction varie suivant leur nature en des séquences de 1 (codes fonctionnels), de 2 (1ère carte BBL de seconde espèce), ou de 3 cartes (autres macro-codes).

2 - Remarques générales

Numérotation des cartes : le compilateur numérote en Mot 2 les cartes qu'il perfore.

Le type de la macro-instruction n'est pas interprété.

Le signe : si une macro-instruction est négative, le compilateur ne tient pas compte du signe -moins- et on ne retrouve nulle trace de celui-ci dans les instructions de la séquence de traduction.

Code opération : le compilateur n'entreprend l'analyse d'une macro-instruction qu'à la condition expresse que ce code opération soit symbolique et appartienne à l'une des 6 tables de Codes. Une macro-instruction dont le code opération est blanc, ou bien numérique, ou symbolique, n'appartenant pas à la liste de 56 codes interprétables, est reproduite telle quelle. Un macro-code opération erroné sera interprété comme le macro-code opération de la liste dont le transcodage sera immédiatement supérieur au sien. Ainsi CMO sera interprété comme CMU.

3 - Remarques (colonnes 63 à 72)

Les remarques de la carte PASO accompagnant une macro-instruction ne sont pas reproduites par le compilateur. Elles sont, en général, remplacées par le macro-code opération et l'adresse facteur de la macro-instruction et ceci dans chacune des cartes de la séquence de traduction.

Afin d'interpréter, s'il y a lieu, la liste des cartes perforées par le compilateur, il est utile de savoir que le compilateur analyse en premier lieu le macro-code opération, puis, en second lieu, les adresses facteur et instruction suivantes en faisant généralement appel à l'instruction suivante pour consultation.

3.31.3 - TRADUCTION DES MACRO - INSTRUCTIONS

1 - Macro - instruction arithmétique

Une macro-instruction arithmétique de la forme :

(A) M x x (F) (I) ,

où F doit être numérique, est traduite par la séquence de 3 cartes

$$\left\{ \begin{array}{l} (A) \quad OO \quad (F) \quad () \\ () \quad RAD \quad (A) \quad () \\ () \quad AAG \quad (I) \quad 17 \frac{50}{95} \end{array} \right.$$

de sorte que , à l'entrée dans l'arithmétique , on a dans 8003 , l'instruction suivante , et dans 8002 l'adresse F en position d'adresse facteur . De plus , la structure de cette séquence permet de retrouver dans la mémoire d'adresse A l'adresse facteur initiale F à sa position normale enchainant avec les 2 autres instructions de la séquence . Cette structure permet la programmation en boucle et toutes les modifications d'instructions portant sur l'adresse facteur .

Si (A) est blanc , il est remplacé par une constante interne symbolique de la forme :

A B 9 9 X

Si (I) est blanc , et que l'instruction suivante est traduite par une séquence de 3 instructions , il est également remplacé par une constante interne symbolique de même forme .

Ces constantes internes symboliques sont construites par le compilateur et sont rendues nécessaires par le fait que dans la traduction de 2 macro-instructions consécutives , l'adresse (A) ou (I) se retrouve dans 3 positions des 2 séquences consécutives et puisque PASO n'accorde le même équivalent numérique qu'à 2 blancs consécutifs , il n'est pas possible de laisser ces adresses en blanc et le Compilateur doit donc leur attribuer le même symbole de la forme A B 9 9 X .

Par exemple , les 2 macro-instructions :

$$\left\{ \begin{array}{l} (A) \quad DZA \quad F \quad () \\ () \quad DAA \quad G \quad () \end{array} \right.$$

seront traduites par les 2 séquences :

$$\left\{ \begin{array}{l} (A) \quad OO \quad (F) \quad () \\ () \quad RAD \quad (A) \quad () \\ () \quad AAG \quad \underline{AB99X} \quad 1764 \end{array} \right.$$

$$\left\{ \begin{array}{l} \underline{AB99X} \quad OO \quad (G) \quad () \\ () \quad RAD \quad \underline{AB99X} \quad () \\ () \quad AAG \quad (AB99W) \quad 1757 \end{array} \right.$$

L'entrée dans l'Arithmétique se fait dans la bande 1750 - 1795 , chacune de ces 46 entrées étant spécifique d'un macro-code symbolique , donc , de l'opération considérée .

Ces entrées sont les suivantes :

CAA : 1750	DAA : 1757	DZA : 1764
CAS : 1751	DAS : 1758	DZS : 1765
CDV : 1752	DAV : 1759	DZV : 1766
CMU : 1753	DDV : 1760	DZW : 1767
CTR : 1754	DMU : 1761	
CZA : 1755	DSV : 1762	
CZS : 1756	DTR : 1763	
GAA : 1768	TAA : 1775	TZA : 1782
GAS : 1769	TAS : 1776	TZS : 1783
GDV : 1770	TAV : 1777	TZV : 1784
GMU : 1771	TDV : 1778	TZW : 1785
GTR : 1772	TMU : 1779	
GZA : 1773	TSV : 1780	
GZS : 1774	TTR : 1781	
et		
VAA : 1786	VDV : 1789	VZA : 1792
VAS : 1787	VMU : 1790	VZS : 1793
VAV : 1788	VSV : 1791	VZV : 1794 et VZW : 1795

2 - Macro-instructions au code en ZZ

Une macro-instruction de la forme :

(A) M Z Z Δ_1 Δ_2

où Δ_1 et Δ_2 sont 2 distances numériques, positives ou nulles, suivant la nature de MZZ, est traduite par la séquence de 3 instructions :

$\left\{ \begin{array}{l} (A) \text{ RAD } A B 9 9 X \text{ () } \\ A B 9 9 X \text{ O O } \Delta_1 \Delta_2 \\ () \text{ A A G (} A_s \text{) } 179 \frac{6}{9} \end{array} \right.$

Comme l'adresse de l'instruction suivante n'est pas indiquée dans la macro-instruction, le compilateur prend automatiquement A_s , l'adresse de la première instruction suivant la macro-instruction en MZZ. Si A_s est blanc, elle sera remplacée par une constante interne à condition que cette instruction suivante soit une macro-instruction non fonctionnelle.

Δ_1 et Δ_2 ont différentes significations suivant la nature de M Z Z . L'entrée dans la bibliothèque se fait en 1796 - 97 - 98 ou 99 suivant le code .

C Z Z : 1796 $\Delta_1 = \Delta$: distance entre partie réelle et imaginaire
 $\Delta_2 = 0000$

D Z Z : 1797 $\Delta_1 = D'$: distance entre Mot 1 et Mot 2
 $\Delta_2 = 0000$

G Z Z : 1798 $\Delta_1 = \Delta$: distance entre partie réelle et imaginaire
 $\Delta_2 = D'$: distance entre les 2 Mots de chaque partie

T Z Z : 1799 $\Delta_1 = D'$: distance entre Mot 1 et Mot 2
 $\Delta_2 = D''$: distance entre Mot 2 et Mot 3

3 - Macro-codes fonctionnels

Une macro-instruction fonctionnelle de la forme :

(A) K F xx (I)

est traduite par la seule instruction :

(A) E D I (I) okkk

• Si I est blanc , il peut être remplacé par une constante interne symbolique à condition que l'instruction suivante soit une macro-instruction non fonctionnelle .

• Si A est blanc , il le reste .

L'entrée dans le sous-programme fonctionnel se fait en okkk ; entrée spécifique de chaque fonction :

KFAT : 0700	KFLE : 0150
KFCN : 0501	KFRC : 0050
KFEX : 0350	KFSN : 0500

4 - Carte BBL de première espèce

Elle est traduite par 1 ou 2 cartes P R A de réservation des zones occupées par la bibliothèque :

BBLA	PRA	1700	1999
BBLB	PRA	1650	1999
BBLC	PRA	1550	1999
BBLD	PRA	1650	1999
BBLE	PRA	1540	1999
BBLF	PRA	1400	1999

BBLG	{ PRA	1700	1999
	{ PRA	0050	0845

BBLH	{ PRA	1650	1999
	{ PRA	0050	0845

BLI	{ PRA	1540	1999
	{ PRA	0050	0845

BBLJ	{ PRA	1400	1999
	{ PRA	0050	0845

Dans les colonnes REMARQUES on retrouve l'indication de la bibliothèque : BBLA par exemple .

5 - Carte BBL de seconde espèce

La première carte est traduite par 2 cartes PRA de réservation de la zone de l'Arithmétique et de la zone de la fonction . Les autres cartes ne sont traduites que par une seule carte PRA de réservation de la zone de la fonction considérée.

- Réservations d'Arithmétique : voir BBLA , B , E et F
- Réservations de fonction :

KFAT	PRA	0645	0844
KFCN	PRA	0480	0644
KFEX	PRA	0310	0479
KFLE	PRA	0150	0309
KFRC	PRA	0050	0149
KFSN	P RA	0480	0644

Dans les colonnes REMARQUES on retrouve la lettre définissant l'Arithmétique et le code fonctionnel utilisé .

6 - Carte Z

Cette carte de fin de compilation se traduit par la carte :

PDT omnp CONSTANTE(S)

indiquant en position d'adresse instruction à suivre , le nombre mnp (<400) de constantes internes symboliques qu'il a été nécessaire de construire pour assurer l'enchaînement des séquences de traduction .

3.32 - 2ème PASSAGE MACHINE : ASSEMBLAGE PASO

La liste compilée peut alors être assemblée par PASO qui perfore un dernier ensemble de

cartes immédiatement utilisables pour entreprendre les calculs .

Se reporter à la brochure PASO .

3.33 - 3ème PASSAGE MACHINE : UTILISATION DU PROGRAMME ASSEMBLE

Le programme assemblé est perforé par PASO en cartes « chargement 1 Mot »

3.33.1 - CHARGEMENT DU TAMBOUR

Les sous-programmes de la bibliothèque ne sont pas nécessairement tous employés et PASO a pu assembler le programme à l'emplacement normal des sous-programmes non utilisés . Pour cette raison :

Enregistrer d'abord sur le tambour la bibliothèque correspondant à l'indicatif fourni au Compileur au début de la compilation .

Enregistrer en second lieu le programme assemblé . Il est alors possible d'entreprendre les calculs.

3.33.2 - ARRETS PROGRAMMES DES BIBLIOTHEQUES

Les bibliothèques contiennent les arrêts programmés suivants :

1 - Arrêts concernant les arithmétiques

01 0111 8000 : Arrêt overflow : l'exposant du résultat de l'opération est supérieur à 99 (à 999 pour la BBL C)

01 0222 8000 : Le sous- programme devrait effectuer une division dont le diviseur est nul ,

01 0333 8000 : L'une ou l'autre des distances Δ , D' ou D'' est nulle alors qu'elle devrait ne pas l'être , ou bien , n'a pas été chargée .

2 - Arrêts concernant les fonctions

01 0021 9999 : Calcul de racine carrée : argument négatif .

01 0022 9999 : Logarithme népérien : argument négatif , nul , ou bien dont l'exposant est en dehors de l'intervalle :] 0,8 et 94 [.

01 0023 9999 : Exponentielle : argument en valeur absolue , égal ou supérieur à 110 .

01 0024 9999 : Sinus et cosinus : argument en valeur absolue , égal ou supérieur à 1000 .

SECTION 3.4

EXEMPLES DE PROGRAMMATION POUR UTILISATION DU COMPILEUR

3.41 - 1er exemple : TABLE DE RACINE CARREE

Le problème reste le même ainsi que la méthode utilisée ; seule diffère l'écriture du programme. Dans le cas présent, nous écrivons les opérations faisant appel à l'Arithmétique Réelle Double Précision au moyen de macro-codes double précision (voir liste) ce qui entraîne quelques adaptations du programme et un triple passage en machine : compilation, assemblage et calculs.

3.41.1 - ECRITURE DU PROGRAMME

Les instructions de calcul Double Précision sont écrites en utilisant les macro-codes en D : (voir liste paragraphe 3.11.2).

Les adresses facteurs de ces macro-instructions doivent être numériques et représenter l'adresse du Mot 1 du nombre traité.

La première carte doit être une carte BBL A.

Avant tout calcul double précision on utilise la macro-instruction

DEP DZZ 0001 0000.

La dernière carte est une carte Z que l'on place après les constantes numériques.

Par contre, les autres instructions standards sont conservées.

TABLE DE RACINE CARREE

		BBL	A		Réservation de bibliothèque
		PRA	0000	1560	
(O) READ		LEC	1950	DEP	
	DEP	DZZ	0001	0000	Chargement de la distance entre Mots
(1) S T A R T		DZA	1951	AB	A_i
(2) AB		DDV	1957		$\frac{A_i}{x_{i,p}}$
		DAA	1957		$A_i/x_{i,p} + x_{i,p}$
		DMU	1550		$(A_i/x_{i,p} + x_{i,p}) \times 0,5 = x_{i,p+1}$
		DTR	1979		$x_{i,p+1} \rightarrow 1979 - 80$
		DAS	1957		$x_{i,p+1} - x_{i,p}$
		ANN	OUI	NON	
OUI		EDI	1979		Nouvelle itération
		TDI	1957		$x_{i,p+1} \rightarrow 1957 - 58$
		EDI	1980		
		TDI	1958	START	(1)
NON		EDI	1979		Calculs complémentaires
		TDI	1957		$x_{i,p+1} \rightarrow 1957 - 58$
		EDI	1980		
		TDI	1958		
		DZA	1552		1
		DDV	1979		$1/\sqrt{A_i}$
		DTR	1981		$1/\sqrt{A_i} \rightarrow 1981 - 82$
		DZA	1979		$\sqrt{A_i}$
		DMU	1979		$(\sqrt{A_i})^2$
		DTR	1983	FIN	$(\sqrt{A_i})^2 \rightarrow 1983 - 84$

FIN	DZA	1951	
	DTR	1977	
	DAS	1955	
	ANN	SUITE	STOP
STOP	PFO	1977	READ
SUITE	DZA	1951	
	DAA	1953	
	DTR	1951	
	PFO	1977	AB
1550	50	5000	0000
1552	51	1000	0000
	Z		

A_i

$A_i \longrightarrow 1977 - 78$

$A_i - A_n$

Perforation dernière carte de la table

(0) Nouvelle Table

A_i

$A_i + b = A_{i+1}$

$A_{i+1} \longrightarrow 1951 - 52$

(2) Perforation du résultat

Constantes numériques

Carte Z de fin de bloc à compiler.

3.41.2 - RESULTATS DE LA COMPILATION

La compilation se fait en suivant les règles du Guide Opérateur et à vitesse de perforation : 100 cartes / minute , chaque carte recevant un numéro .

La carte BBL Aest traduite par la réservation :

PRA 1700 1999 BBL A

La macro-instruction DEP DZZ 0001 0000 est traduite par la séquence :

{	DEP	RAD	02999	()	DZZ 0001
	02999	00	0001	0000	DZZ 0001
	()	AAG	START	1797	DZZ 0001

Les macro-instructions arithmétiques sont traduites par des séquences de 3 instructions ;

ainsi , () DAA 1957 ()
 devient : { 02998 00 1957 () DAA 1957
 () RAD 02998 () DAA 1957
 () AAG 02997 1757 DAA 1957

Les instructions PASO standards , les pseudo-instructions et les constantes numériques sont reproduites telles quelles .

La carte Z est traduite par :

PDT 0015 . CONSTANTES

signifiant que 15 constantes de la forme 02999 , 02998 ont été nécessaires à l'enchaînement des séquences de traduction .

La liste initiale du programme comportait 37 instructions , pseudo-instructions , constantes dont 19 macro-instructions arithmétiques ; la liste traduite en comportera donc :

$$(19 \times 3) + 18 = 75 \text{ cartes}$$

Le programme se trouve donc allongé dans la mesure où il comporte des macro-instructions.

3.41.3 - UTILISATION et RESULTATS

La liste de traduction est alors assemblée en utilisant PASO III - Voir liste -

L'utilisation du programme et les résultats des calculs restent les mêmes que dans le cas d'une écriture directe du programme .

[illegible]

ENTRÉE

A ₁	A ₂	pour h		A ₃	Z ₁₀	VERIFI	CATION
		RACINE	CARREE				
5111000000	0000000000	5010000000	0000000000	5131000000	0000000000	5110200000	0000000000
5111000000	0000000000	5110480088	4817015153	5095346258	9245592313	5111000000	0000000001
5112000000	0000000000	5110954451	1501033223	5091287092	9175276853	5112000000	0000000001
5113000000	0000000000	5111401754	2509913798	5087705801	9307029214	5113000000	0000000000
5114000000	0000000000	5111832159	5661992321	5084515425	4728516576	5114000000	0000000000
5115000000	0000000000	5112247448	7139158905	5081649658	0927726032	5115000000	0000000000
5116000000	0000000000	5112649110	6406735174	5079058941	5042094829	5116000000	0000000002
5117000000	0000000000	5113036404	8104052975	5076696496	8847370433	5117000000	0000000002
5118000000	0000000000	5113416407	8649987382	5074535599	2499929898	5118000000	0000000003
5119000000	0000000000	5113784048	7520902218	5072547625	0110011670	5119000000	0000000001
5120000000	0000000000	5114142135	6237309505	5070710678	1186547524	5120000000	0000000000
5121000000	0000000000	5114491376	7461824386	5069006555	9342354217	5121000000	0000000001
5122000000	0000000000	5114832396	9741913259	5067419986	2463242086	5122000000	0000000000
5123000000	0000000000	5115165750	8881031011	5065938047	3395787005	5123000000	0000000000
5124000000	0000000000	5115491933	3843296676	5064549722	4367902812	5124000000	0000000002
5125000000	0000000000	5115811388	3008418957	5063245553	2033675865	5125000000	0000000001
5126000000	0000000000	5116124515	4965970993	5062017367	2946042281	5126000000	0000000000
5127000000	0000000000	5116431676	7251549834	5060858061	9450184571	5127000000	0000000000
5128000000	0000000000	5116733200	5306815110	5059761430	4667196819	5128000000	0000000001
5129000000	0000000000	5117029386	3659264012	5058722021	9514703487	5129000000	0000000001
5130000000	0000000000	5117320508	0756887730	5057735026	9189625762	5130000000	0000000002
5131000000	0000000000	5117606516	8616590092	5056796183	4247064809	5131000000	0000000002



138

A ₁	pas h	RACINE		A _n	Z _{n,0}	VERIFI	
		CARREE	INVERSE DE RACINE			CATION	
5120000000	0000000000	5114142135	6237309505	5070710678	-1186547524	5120000000	0000000000
5130000000	0000000000	5117320508	0756887730	5057735026	9189625762	5130000000	0000000002
5140000000	0000000000	5120000000	0000000000	5050000000	0000000000	5140000000	0000000000
5150000000	0000000000	5122360679	7749978970	5044721359	549957939	5150000000	0000000002
5160000000	0000000000	5124494897	4278317810	5040924829	0463863016	5160000000	0000000001
5170000000	0000000000	5126457513	1106459059	5037796447	3009227227	5170000000	0000000000
5180000000	0000000000	5128284271	2474619010	5035355339	0593273762	5180000000	0000000001
5190000000	0000000000	5130000000	0000000000	5033333333	3333333333	5190000000	0000000000
5200000000	0000000000	5131622776	6016837933	5031622776	6016837933	5199999999	9999999995
5210000000	0000000000	5133156247	9035539985	5030151134	4577763623	5210000000	0000000000
5220000000	0000000000	5134641016	1513775455	5028867513	4594812892	5212000000	0000000000
5230000000	0000000000	5136055512	7546398930	5027735009	8112614560	5213000000	0000000001
5240000000	0000000000	5137416573	8677394139	5026725124	1912424384	5214000000	0000000000
5250000000	0000000000	5138729833	4620741689	5025819888	9747161125	5215000000	0000000000
5260000000	0000000000	5140000000	0000000000	5025000000	0000000000	5215000000	0000000000
5270000000	0000000000	5141231055	2561766055	5024253562	50363332974	5217000000	0000000000
5280000000	0000000000	5142426406	8711928515	5023570226	0395515841	5218000000	0000000000
5290000000	0000000000	5143588989	4354067355	5022941573	3870561756	5219000000	0000000000
5300000000	0000000000	5144721359	5499957940	5022360679	7749978969	5220000000	0000000001
5310000000	0000000000	5145825756	9495584001	5021821789	0235992391	5221000000	0000000000

3.42 - 2ème exemple : CALCUL de e^z

Le problème reste le même ainsi que la méthode de calcul utilisée ; seule diffère l'écriture du programme . Dans le cas présent, nous écrivons les opérations faisant appel à l'Arithmétique Complexe Double Précision au moyen de macro-codes G, double précision complexe , ce qui entraîne quelques adaptations dans la réinitialisation du programme , dans les tests de fin de boucle et un triple passage machine : compilation , assemblage et calculs .

3.42.1 - ECRITURE DU PROGRAMME

Les instructions de calcul Double Précision Complexe sont écrites en utilisant les macro-codes en G (voir liste paragraphe 3.11.2) .

Les adresses facteurs de ces macro-instructions sont numériques et représentent l'adresse du Mot 1 de la partie réelle des nombres traités .

La première carte doit être une carte BBL E.

Avant tout calcul complexe , on utilise la macro-instruction

GZZ 0002 0001

La dernière carte est une carte Z que l'on place après les constantes numériques .

Par contre, les autres instructions standards sont conservées (aux adaptations près).

	BBL	E	
	PRA	0000	0750
	PRA	0930	1350
ERSTE	RAD	Z	8002
AZO	SAD	ZL	
	ANN		VU
	AAD	8001	
	AAD	UN	8002
VU	RAD	LA	
	EDI	BERGN	
	SBF	BERGN	
	RAD	LD	
	EDI	BZZ	
	SBF	BZZ	
	GZZ	0002	0001
	RAD	2	
	TRD	0930	
	TRG	0931	
	TDI	0932	
	TDI	0933	READ

Réserve de bibliothèque

*Réinitialisation de la zone de
stockage des U_p
de 0950 à 1350*

*Réinitialisation de l'instruction
d'appel de U_p départ en 0954*

*Réinitialisation de l'instruction
de transfert de U_{p+1}
Départ en 0958*

*Transfert des distances entre
Mots*

*Réinitialisation de diviseur $p+1$
dont la valeur est +2*

READ	LEC	1950	START
START	EDI	1951	
	TDI	1977	
	TDI	0954	
	EDI	1952	
	TDI	1978	
	TDI	0955	
	EDI	1953	
	TDI	1979	
	TDI	0956	
	EDI	1954	
	TDI	1980	
	TDI	0957	BERGN
<u>BERGN</u>	GZA	0954	
	GMU	0951	
	GDV	0930	BZZ
BZZ	GTR	0958	TST

Lecture de la valeur de Z

*Z qui est aussi U_1
est transféré sur mémoires de
sortie - 1977 - 79 - 80 - 81*

*et sur la zone de stockage
en 0954 - 55 - 56 - 57*

Calcul de $U_{p+1} U_p : (0954+4p)$

$$U_p \times Z$$

$$U_p \times \frac{Z}{p+1} = U_{p+1}$$

Transfert de U_{p+1} en 0958 + 4p

TST	EDI	1650	NON
	TDI	0934	
	EDI	1665	
	TDI	0935	
	RSB	1655	
	SAD	1670	
	TRG	0936	
	TRD	0937	
	GMU	0934	
	GAS	0938	
	RAG	1650	
	AAD	1665	
	ANG	OUI	
NON	RAD	BZZ	BERGN
	EDI	BERGN	
	SBF	BERGN	
	AAD	4	
	TRD	BZZ	
	GZA	0930	
	GAA	0950	
	GTR	0930	

Transfert du conjugué de U_{p+1}
dans les mémoires
0934 - 35 - 36 - 37

Produit $U_{p+1} \times \overline{U_{p+1}} = |U_{p+1}|^2$

$|U_{p+1}|^2 - \varepsilon^2$ est un nombre réel
Transfert du résultat de cette
différence dans 8003 - 8002

Préparation au calcul de U_{p+2}

Modification de l'instruction
d'appel de
 $U_{p+1} (p+1 \rightarrow p+2)$

Modification de l'instruction
de transfert de
 $U_{p+1} (p+1 \rightarrow p+2)$
sur la zone de stockage
 $p+1$

$(p+1) + 1 = p+2$
 $p+2$ transféré en 0930 -1 - 2 -3

OUI	RAD	BZZ	
	EDI	BXX	
	SBF	BXX	
	SAD	4	
	EDI	BTT	
	SBF	BTT	BXX
BXX	GZA	9999	BTT
<u>BTT</u>	GAA	9999	
	RAD	BTT	
	AAG	8003	
	SBF	BUU	
	RAD	8001	
	SAD	LI M	
	ANN		FIN
	AAD	8001	
	SAD	4	
	EDI	BTT	
	SBF	<u>BTT</u>	<u>8001</u>

Préparation de la 1ère instruction de sommation des U_p , depuis U_n à U_o

Instruction d'addition de U_k pour $k = n-1, \dots, 0$

Appel de U_n

Addition de U_k

L'adresse du U_k additionnée est-elle 0950 ?

Puisque U_k n'est pas U_o , on diminue l'adresse de 4 unités pour additionner U_{k-1}

FIN	GTR	1981	
	PFO	1977	
	RAD	BZZ	
	AAG	8003	
	SBF	B UU	
	RAG	8001	
	SAG	LIM	
	DIV	44	
	DAD	0001	
	EDI	STOP	
	SBF	STOP	8001
STOP	ART	9999	8000

Transfert de la somme
 $\sum_{k=n}^0 U_k$ sur mémoire de perfo
 1981 - 2 - 3 - 4

Perforation du résultat

Calcul du nombre de termes
 calculés

que l'on fait apparaître au pupitre

Z	TRG	0954	AZO
ZL	TRG	1350	AZO
UN	00	0001	0000
LA	00	0954	0000
LD	00	0958	0000
4	00	0004	0000
44	40	0000	0000
LIM	00	0950	0000
9 50	51	1000	0000
938	15	2000	0000
2	51	2000	0000
	Z		

Constantes

Nombre réel + 1 : Mot 1

Nombre réel ε^2 : Mot 1

Nombre réel + 2 : Mot 1

Carte Z de fin de bloc à
 compiler

3.42.2 - RESULTATS DE LA COMPILATION

La compilation se fait en suivant les règles du Guide Opérateur et à vitesse de perforation : 100 cartes / minute , chaque carte recevant un numéro .

La carte BBL E est traduite par la réservation :

PRA 1540 1999 BBL E

La macro-instruction GZZ 0002 0001 est traduite par la séquence :

$$\left\{ \begin{array}{l} 02999 \text{ RAD } 02998 \text{ () GZZ 0002} \\ 02998 \text{ 00 0002 0001 GZZ 0002} \\ \text{() AAG () 1798 GZZ 0002} \end{array} \right.$$

Comme l'instruction suivante est une instruction standard, il n'est pas nécessaire d'attribuer une constante interne à (I) qui reste blanc .

Les macro-instructions arithmétiques sont traduites par des séquences de 3 instructions . Ainsi :

BERGN GZA 0954 ()

devient :

$$\left\{ \begin{array}{l} \text{BERGN 00 0954 () GZA 0954} \\ \text{() RAD BERGN() GZA 0954} \\ \text{() AAG 02997 1773 GZA 0954} \end{array} \right.$$

Les instructions PASO standards , les pseudo-instructions et les constantes numériques sont reproduites telles quelles .

La carte Z est traduite par :

PDT 0009 CONSTANTES

signifiant que 9 constantes de la forme 0299 X ont été nécessaires à l'enchaînement des séquences de traduction .

La liste initiale du programme comportait 100 instructions ou pseudo-instructions , donc 13 macro-instructions arithmétiques ; la liste traduite en comportera donc :

$$(13 \times 3) + 87 = 126 \text{ cartes}$$

Le programme se trouve donc allongé dans la mesure où il comporte des macro-instructions .

3.42.3 - UTILISATION et RESULTATS

La liste de traduction est alors assemblée en utilisant PASO III - Voir liste .

L'utilisation du programme et les résultats des calculs restent les mêmes que dans le cas d'une écriture directe du programme .



1	0800	55	0753	0002	PRA	1540	1999	ABLE	PSEUDO
2	0850	15	0760	0757	PRA	0000	0750		PSEUDO
3	0750	45	0761	0761	PRA	0930	1350		PSEUDO
4	0760	15	0772	0769	SAD	ZL	8002		
5	0769	15	0772	8002	ANN	8001	VU		
6	0769	15	0772	8002	AAD	UN	8002		
7	0769	15	0772	8002	AAD	UN	8002		
8	0769	15	0772	8002	AAD	UN	8002		
9	0769	15	0772	8002	AAD	UN	8002		
10	0769	15	0772	8002	AAD	UN	8002		
11	0769	15	0772	8002	AAD	UN	8002		
12	0769	15	0772	8002	AAD	UN	8002		
13	0769	15	0772	8002	AAD	UN	8002		
14	0769	15	0772	8002	AAD	UN	8002		
15	0769	15	0772	8002	AAD	UN	8002		
16	0769	15	0772	8002	AAD	UN	8002		
17	0769	15	0772	8002	AAD	UN	8002		
18	0769	15	0772	8002	AAD	UN	8002		
19	0769	15	0772	8002	AAD	UN	8002		
20	0769	15	0772	8002	AAD	UN	8002		
21	0769	15	0772	8002	AAD	UN	8002		
22	0769	15	0772	8002	AAD	UN	8002		
23	0769	15	0772	8002	AAD	UN	8002		
24	0769	15	0772	8002	AAD	UN	8002		
25	0769	15	0772	8002	AAD	UN	8002		
26	0769	15	0772	8002	AAD	UN	8002		
27	0769	15	0772	8002	AAD	UN	8002		
28	0769	15	0772	8002	AAD	UN	8002		
29	0769	15	0772	8002	AAD	UN	8002		
30	0769	15	0772	8002	AAD	UN	8002		
31	0769	15	0772	8002	AAD	UN	8002		
32	0769	15	0772	8002	AAD	UN	8002		
33	0769	15	0772	8002	AAD	UN	8002		
34	0769	15	0772	8002	AAD	UN	8002		
35	0769	15	0772	8002	AAD	UN	8002		
36	0769	15	0772	8002	AAD	UN	8002		
37	0769	15	0772	8002	AAD	UN	8002		
38	0769	15	0772	8002	AAD	UN	8002		
39	0769	15	0772	8002	AAD	UN	8002		
40	0769	15	0772	8002	AAD	UN	8002		
41	0769	15	0772	8002	AAD	UN	8002		
42	0769	15	0772	8002	AAD	UN	8002		
43	0769	15	0772	8002	AAD	UN	8002		
44	0769	15	0772	8002	AAD	UN	8002		
45	0769	15	0772	8002	AAD	UN	8002		
46	0769	15	0772	8002	AAD	UN	8002		
47	0769	15	0772	8002	AAD	UN	8002		
48	0769	15	0772	8002	AAD	UN	8002		
49	0769	15	0772	8002	AAD	UN	8002		
50	0769	15	0772	8002	AAD	UN	8002		



	Z	E	DOUBLE	PRECISION	PARTIE	REELLE	PARTIE	IMAGINAIRE
5110000000	0000000000	0000000000	0000000000	0000000000	5127182818	2845904524	0000000000	0000000000
5110000000	0000000000	5110000000	0000000000	0000000000	5114686939	3991588516	5122873552	8717884239
5110000000	0000000000	5112000000	0000000000	0000000000	5098499049	9395166040	5125335449	1079828929
5110000000	0000000000	5180000000	0000000000	0000000000	-5039551009	-7942129300	5126893545	4363244170
5210000000	0000000000	5210000000	0000000000	0000000000	-5518481780	-3345986488	-5511982862	-3906574560

3. 43 - 3ème exemple : CALCUL DIRECT de $e^x \cos y$ et de $e^x \sin y$

Connaissant x et y , on calcule directement partie réelle et partie imaginaire de e^{x+iy} à l'aide des fonctions double précision : cosinus, sinus et exponentielle.

Ce programme, dont la réalisation ne présente aucune difficulté, est essentiellement utilisé dans le but d'illustrer l'emploi des codes fonctionnels et de vérifier les calculs précédents sur e^z .

343.1 - ECRITURE DU PROGRAMME

Les instructions de calcul double précision sur nombres réels, sont écrites en utilisant les macro codes D et les instructions fonctionnelles en utilisant les macro-codes K nécessaires.

Les adresses facteurs de ces macro-instructions sont numériques et représentent l'adresse du Mot 1 du nombre traité.

Comme l'on utilise uniquement des codes D (BBL A) et les fonctions cosinus, exponentielle et sinus, les 3 premières cartes seront des cartes BBL A de seconde espèce.

{	BBL A	KFCN
	BBL A	KFEX
	BBL A	KFSN

Avant tout calcul, on utilise la macro-instruction :

DZZ 0001 0000

La dernière carte de la liste est une carte Z.

Avant d'utiliser un code fonctionnel, il faut charger l'argument en 8003-2, ce que réalise l'utilisation d'un code DZA. Mais alors il n'est pas possible d'utiliser un code DTR pour transférer x et y sur mémoire de perforation puisque 8003-2 se trouve modifié.

Ces instructions sont renvoyées en fin de programme.

Voir liste du macro-programme.

<u>READ</u> GO	BBL	A	KFCN
	BBL	A	KFEX
	BBL	A	KFSN
	PRA	0000	0850
	PRA	0948	1999
	LEC	1950	GO
	DZZ	0001	0000
	DZA	001	
	KFE	X	
	DTR	0948	
	IZA	053	
	KFC	N	
	DMU	0948	
	DTR	1981	
	DZA	053	
	KFS	N	
	DMU	0948	
	DTR	1983	
	DZA	1951	
	DTR	1977	
	DZA	1953	
	DTR	1979	
	PFO	1977	<u>READ</u>
	Z		

*Réservation de bibliothèque
et de zones de fonction*

Lecture en 1951 - 1960

Transfert de distance entre Mots

$$(8003 - 2) = x$$

$$(8003 - 2) = e^x$$

Transfert e^x en 0948 - 0949

$$(8003 - 2) = y$$

$$(8003 - 2) = \cos y$$

$$(PA) = \cos y \times e^x$$

*$e^x \cos y$ transféré sur zone
perfo*

$$(8003 - 2) = y$$

$$(8003 - 2) = \sin y$$

$$(PA) = \sin y \times e^x$$

*$e^x \sin y$ transféré sur zone
perfo*

x transféré sur zone perfo

y transféré sur zone perfo

Perforation des résultats

Renvoi à une nouvelle carte

3.43.2 • RESULTATS DE COMPILATION

La compilation se fait suivant les règles du Guide Opérateur et à vitesse de perforation :
100 cartes par minute , chaque carte recevant un numéro .

La première carte BBL A KFCN est traduite par :

{	PRA	1700	1999		A KFCN
	PRA	0480	0644		A KFCN

et les deux autres par :

PRA	0310	0479		A KFEX
PRA	0480	0644		A KFSN

La macro-instruction GO DZZ 0001 0000 devient :

{	GO	RAD	02999	()	DZZ 0001
	02999	00	0001	0000	DZZ 0001
	()	AAG	02998	1797	DZZ 0001

Comme l'instruction suivante est traduite par une séquence de 3 cartes et que son adresse A est blanche , il est nécessaire de la remplacer par la constante interne 02998 .

Les macro-instructions arithmétiques sont traduites par des séquences de 3 cartes .

Ainsi () DTR 1977 ()

devient {	0299 X	00	1977	()	DTR 1977
	()	RAD	0299 X	()	DTR 1977
	()	AAG	0299 W	1763	DTR 1977

Les macro-instructions fonctionnelles sont traduites par une seule carte .

Par exemple :

KFE X

devient	EDI	02997	0350	K FE X
---------	-----	-------	------	--------

Les instructions standards , les pseudo-instructions PASO sont reproduites telles quelles.

La carte Z est traduite par la

PDT 0013 CONSTANTES

signifiant que 13 constantes de la forme 0299X ... ont été nécessaires à l'enchainement des séquences de traduction .

La liste initiale du programme comportait 24 instructions ou pseudo-instructions ; la liste traduite en comporte 51 .

3.43.3 - UTILISATION

1 - Caractéristique du programme - Il se présente sous la forme d'un paquet de 52 cartes 1 Mot avec carte départ calcul automatique en 0900 .

Le programme occupe la zone de 0851 à 0949 avec 53 mémoires de libres .

2 - Format des cartes données - La zone de lecture est celle de la bande 1950 , la carte étant non - chargement .

Mots 1 et 2 : x en double précision

Mots 3 et 4 : y en double précision

Mots 5 à 8 : 0000000000 +

3 - Mode d'emploi - Placer dans le magasin d'alimentation et dans l'ordre la BBL A (avec son programme de chargement) , la bibliothèque fonctionnelle et le programme de calcul et les cartes données .

Inscrire 70 0004 xxxx + au pupitre pour la remise à zéro du tambour .

Appuyer sur REST - GENE , DEP , PROG et DEP. LECTURE .

Après la remise à zéro , inscrire 70 1951 xxxx + au pupitre et appuyer sur DEP . PROG .

Après chargement , le calcul démarre automatiquement en 0900 .

En fin de calcul , le programme s'arrête sur l'instruction :

70 1950 0851

3.43.4 - RESULTATS

1 - Format des cartes sortie - La zone de sortie est celle de la bande 1950 : (1977 - 86)

Mots 1 et 2 : x en double précision

Mots 3 et 4 : y en double précision

Mots 5 et 6 : $e^x \cos y$ en double précision

Mots 7 et 8 : $e^x \sin y$ en double précision

2 - Temps de calcul - En moyenne , le temps de calcul est de l'ordre de 5 à 6 secondes .

3 - Précision du résultat - Des calculs effectués en triple précision permettent de contrôler l'exactitude des résultats .

pour $x = 1$ et $y = 1,2$, il y a une erreur de 2 unités du dernier ordre sur $e^x \cos y$.,

pour $x = 1$ et $y = 8$, elle atteint 13 unités ,

ce qui reste très convenable .

4 - Comparaison avec l'exemple N° 2 -

Les méthodes étant totalement différentes et le volume des calculs sans commune mesure ,
il n'y a aucune comparaison possible entre les 2 programmes .

Le temps de calcul est réduit dans de très grandes proportions (95 à 99 % /_o).

La précision est très accrue du fait du volume réduit des calculs .



PSEUDO
PSEUDO
PSEUDO
PSEUDO
PSEUDO
PSEUDO

1	0851	70	1950	0851	READ	PRA	1700	1999	A	DZ	0001	KFON
2	0854	65	0854	0859	GO	LEC	0480	0644	A	DZ	0001	KFON
3	0859	10	0862	1797	02999	00	0310	0479	A	DZ	0001	KFON
4	0862	10	1951	0866	02998	AAG	0480	0644	A	DZ	1951	KFEX
5	0866	65	0862	0867	02998	00	0000	0850		DZA	1951	KFEX
6	0867	10	0870	1764	02998	RAD	0948	1999		DZA	1951	KFEX
7	0870	10	0873	0350	02998	AAG	1950	GO		DZA	1951	KFEX
8	0873	69	0873	0877	02997	EDI	02999			DZ	0001	
9	0877	65	0873	0927	02997	00	0948	0000		DZ	0001	
10	0877	65	0873	0927	02997	RAD	02997	1797		DZ	0001	
11	0877	10	0880	1763	02996	AAG	02996	0000		DZ	1951	
12	0880	10	0880	0884	02996	00	1953	0850		DZ	1951	
13	0884	65	0880	0885	02996	RAD	02996	1999		DZ	1951	
14	0885	10	0886	1764	02996	AAG	02996	GO		DZ	1951	
15	0885	69	0886	1764	02996	EDI	02996			DZ	1951	
16	0885	65	0886	0895	02995	00	0948	0000		DZ	1951	
17	0885	65	0886	0895	02995	RAD	02995	0000		DZ	1951	
18	0885	10	0886	1761	02994	AAG	02994	1797		DZ	1951	
19	0885	10	0886	0852	02994	00	1981	0850		DZ	1951	
20	0885	65	0886	0853	02993	RAD	02993	0850		DZ	1951	
21	0885	10	0886	1763	02993	AAG	02993	1763		DZ	1951	
22	0885	65	0886	0860	02993	00	1953	0850		DZ	1951	
23	0885	10	0886	0861	02993	RAD	02993	1764		DZ	1951	
24	0885	69	0886	1764	02992	EDI	02992	0500		DZ	1951	
25	0885	65	0886	0871	02992	00	0948	0500		DZ	1951	
26	0885	10	0886	0871	02992	RAD	02992	1761		DZ	1951	
27	0885	10	0886	1761	02992	AAG	02992	1761		DZ	1951	
28	0885	65	0886	0878	02992	RAD	02992	1763		DZ	1951	
29	0885	10	0886	0879	02992	00	1983	1763		DZ	1951	
30	0885	65	0886	1763	02992	AAG	02992	1763		DZ	1951	
31	0885	10	0886	0867	02992	RAD	02992	1763		DZ	1951	
32	0885	65	0886	0867	02992	00	1951	1763		DZ	1951	
33	0885	10	0886	1764	02992	AAG	02992	1763		DZ	1951	
34	0885	65	0886	0894	02992	RAD	02992	1763		DZ	1951	
35	0885	10	0886	0894	02992	00	1977	1763		DZ	1951	
36	0885	65	0886	0896	02992	AAG	02992	1763		DZ	1951	
37	0885	10	0886	1763	02992	RAD	02992	1763		DZ	1951	
38	0885	65	0886	0903	02992	00	1953	1763		DZ	1951	
39	0885	10	0886	0904	02992	RAD	02992	1763		DZ	1951	
40	0885	65	0886	1764	02992	AAG	02992	1763		DZ	1951	
41	0885	10	0886	0911	02992	00	1979	1763		DZ	1951	
42	0885	65	0886	0912	02992	RAD	02992	1763		DZ	1951	
43	0885	10	0886	1763	02992	AAG	02992	1763		DZ	1951	
44	0885	65	0886	0903	02992	PFO	1977	1763		DZ	1951	
45	0885	10	0886	0904	02992			1763		DZ	1951	
46	0885	65	0886	1764	02992			1763		DZ	1951	
47	0885	10	0886	0911	02992			1763		DZ	1951	
48	0885	65	0886	0912	02992			1763		DZ	1951	
49	0885	10	0886	1763	02992			1763		DZ	1951	
50	0885	71	1977	0900	02992			1763		DZ	1951	



Z	EN	DOUBLE	PRECISION	PARTIE	REELLE	PARTIE	IMAGINAIRE
5110000000	0000000000	0000000000	0000000000	5127192518	2845904524	0000000000	0000000000
5110731000	0000000000	5053640000	0000000000	5125107125	0457402631	5114995401	1579717431
5110000000	0000000000	5110000000	0000000000	5114555934	3991508516	5122273512	6717004239
5110000000	0000000000	5112000000	0000000000	5056430047	0395156040	5125335449	1079828329
5110000000	0000000000	5115707963	2675489441	5427140510	0645004504	5127132818	3945904124
5110000000	0000000000	5118000000	0000000000	-5061759932	-5332044061	5126471923	1870408973
5120000000	0000000000	5110000000	0000000000	5139523240	4844127144	5116217673	1236756817
5110000000	0000000000	5120000000	0000000000	-5039551009	-7542130030	5126593545	4363244124
5210000000	0000000000	5210000000	0000000000	-5518491780	-3345005420	-5111082862	-3906574153

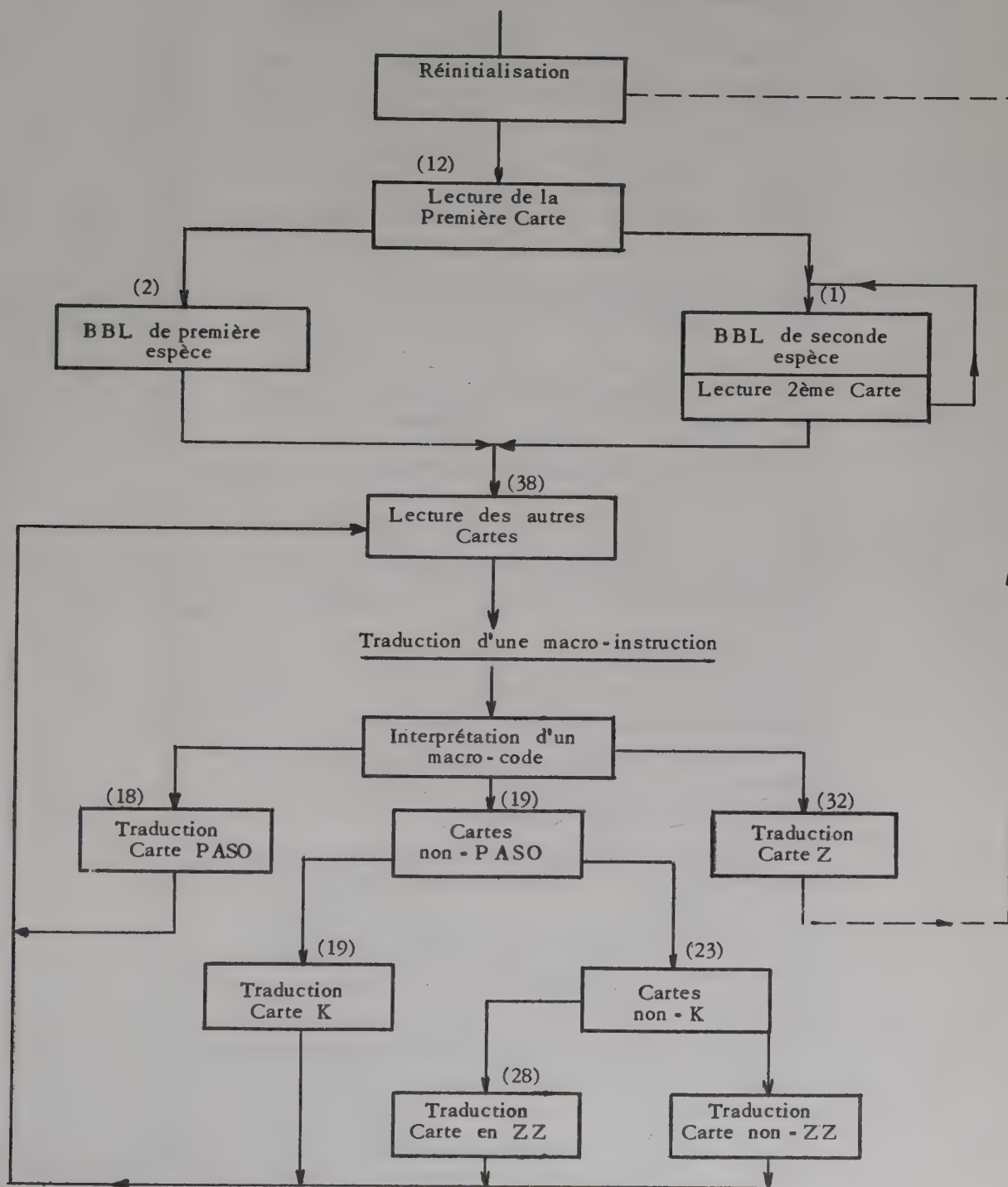
CHAPITRE IV

Etude du Compileur

SECTION 4.1

TABEAU SYNOPTIQUE DE LA COMPILATION D'UN PROGRAMME

4.11 - ORGANIGRAMME GENERAL DU COMPILEUR



4.12 - TABLEAU SYNOPTIQUE DE LA COMPILATION

TRADUCTION D'UNE CARTE BBL DE 1ère ESPECE

- Elimination des cartes chargement et des cartes non-BBL
- Séparation des cartes BBL de seconde espèce par test sur le contenu des colonnes 57 à 61
- Sélection de la bibliothèque par consultation de la table des indicatifs de bibliothèque
- Préparation des réservations des zones (1 ou 2) correspondant à la bibliothèque
- Libération des mémoires d'incompatibilité
- Numérotation et perforation de :

	PRA	1xxx	1999	BBL X	
ou	{	PRA	0050	0845	BBL X
		PRA	1xxx	1999	BBL X

TRADUCTION D'UNE CARTE BBL DE 2ème ESPECE

- Adaptation du programme : modification de la recherche des incompatibilités pour codes K et de la sortie pour une carte BBL de 1ère espèce
- Recherche de la bibliothèque arithmétique A , B , E ou F
- Perforation de la réservation correspondante
PRA 1xxx 1999 X KFxx
- Recherche de la fonction par consultation de la Table K
- Préparation de la réservation de la zone correspondant à la fonction KFxx
- Libération des mémoires d'incompatibilité correspondantes
- Numérotation et perforation de
PRA 0..... 0..... X KFxx
- Lecture de la carte suivante - Orientation vers l'interprétation du codé ou vers la traduction de la carte BBL de seconde espèce suivante
- Vérification de l'indicatif de bibliothèque.

INTERPRETATION D'UN MACRO-CODE Mxx

- Elimination des cartes « chargement » et BBL
- Séparation de la carte Z
- Recherche de la première lettre du code par consultation de la Table des premières lettres
 - Construction de l'instruction d'appel de la mémoire d'incompatibilité
 - Première séparation des codes standards PASO
 - Choix de la Table à consulter
- Seconde recherche du code
 - Cas des codes V
 - Cas des codes K avec BBL de 1ère espèce
 - Cas des codes K avec BBL de 2ème espèce
 - Cas des autres codes : C , D , G et T
- Seconde séparation des codes standards PASO
 - Appel de l'adresse d'entrée dans la bibliothèque correspondant au code ,
 - Deuxième séparation des codes standards PASO
 - Tests sur la mémoire d'incompatibilité
- Séparation des codes MZZ des codes Non ZZ
- Orientation entre codes PASO et Non PASO ; entre continuation de la traduction et consultation de la carte suivante

TRADUCTION D'UNE CARTE PASO

- Libérations des mémoires de la zone de lecture . Séparation entre codes PASO et codes Non-PASO (test en 8/9)
- Cas où F est blanc et I non-blanc : consultation de la carte suivante :
 - Code Non-PASO , non K : construction d'une constante symbolique interne AB99X
 - Code PASO ou K : F reste blanc
 - Perforation de
 - (A) COP { (AB99X) (I)
- Cas où F est non blanc :
 - si I non blanc , perforation de la carte

- si I est blanc , consultation de la carte suivante:
- Code non PASO , non K , construction d'une constante symbolique interne AB 99 W
- Code PASO ou K : perforation sans modification:
I reste blanc

(A) COP (F) ()
(AB99W)

TRADUCTION D'UNE CARTE K

- Séparation des codes K des codes non K
- Préparation des zones de sortie
- Cas où l'adresse I est blanche : consultation de la carte suivante :
 - Code non PASO et non K : construction de constante symbolique interne AB99 V
 - Code PASO ou K : I reste blanc
- Cas où l'adresse I est non blanche : aucune modification
- Numérotation et perforation de la carte :

(A) EDI (I) Okkk KFxx
AB99V

PARTIE COMMUNE AUX CODES NON K

- Transfert du code opération et de F en colonnes REMARQUES des zones de perforation intéressées
- Transferts préliminaires des quantités déjà connues -
Séparation des codes MZZ des codes non ZZ

TRADUCTION D'UNE CARTE NON ZZ

- Numérotation et perforation de la première carte

(A) oo (F) () Mxx (F)

- Numérotation et perforation de la seconde carte :

() RAD (A) () Mxx (F)

- Préparation pour la 3ème carte :

- si I est blanc , consulter la carte suivante :
- Code non PASO , non K : construction d'une constante symbolique interne AB99S
- Code PASO ou K : I reste blanc
- si I est non blanc : aucune modification

- Numérotation et perforation de la 3ème carte :

() AAG (I) 17xx Mxx (F)
AB99S

TRADUCTION D'UNE CARTE MZZ

- Construction de la constante AB99T

- Numérotation et perforation de la 1ère carte :

(A) RAD AB99T () MZZ Δ_1

- Numérotation et perforation de la seconde carte :

AB99T OO Δ_1 Δ_2 MZZ Δ_1

- Préparation pour la 3ème carte :

consultation obligatoire de la carte suivante :

- Cas où A_s est blanc : s'il s'agit d'un code non-PASO, non K , construire la constante symbolique interne

AB99S

- Cas où A_s est non blanc , on l'utilise à la place de I .

- Numérotation et perforation de la 3ème carte :

() AAG AB99S 179x MZZ Δ_1
(A_s)

TRADUCTION DE LA CARTE Z

- Transcodage du nombre omnp du nombre de constantes symboliques internes utilisées

- Test sur ce nombre : singulier ou pluriel
- Numérotation et perforation de la carte

PDT omnp CONSTANTE (S)

SECTION 4.2

REMARQUES CONCERNANT LE COMPILEUR

4.21 - TABLE DES CODES

Le Compileur est accompagné de 6 tables de constantes formées des transcodages des 6 classes de macro-code opération , et dont l'utilisation concerne isolément ou concurremment les différentes bibliothèques .

Ce sont : les Tables.... C , D , G , K , T et V

Chaque table contient tous les macro-codes opération classés par ordre alphabétique , dont le Compileur doit disposer pour interpréter une macro-instruction et élaborer la séquence de traduction qui programme l'entrée dans la bibliothèque choisie .

Mais comme à l'intérieur de chacune de ces tables , la recherche des macro-codes se fait par consultation de table , il est nécessaire que la table consultée comprenne également tous les codes PASO standards commençant par la première lettre de la table.. Ainsi , la Table D contient les 12 macro-codes D et les codes DAD , DDA , DAG , DIV et DRR . Ces 17 codes classés par ordre alphabétique constituant la Table D .

Avant d'interpréter un macro-code , le Compileur exécute une présélection portant sur la 1ère lettre et permettant d'éliminer ainsi tous les codes ne correspondant pas à l'une des 6 Tables , et par le fait même , de choisir la Table à l'intérieur de laquelle devra se faire la consultation .

A la fin de cette sélection , un contrôle a lieu dans le but de vérifier si le code interprété et reconnu dans la Table est compatible avec la bibliothèque indiquée . Ces contrôles se font sur le contenu des mémoires d'incompatibilité , libérées ou non , lors de l'analyse de la PSEUDO-INSTRUCTION BBL lue en première carte . Chaque BBL libère les mémoires d'incompatibilité des différentes Tables correspondant aux macro-codes qu'elle peut utiliser .

Voir la liste des 6 Tables en début de l'Annexe 7 de l'analyse du Compileur .

Les codes soulignés sont des codes standards PASO .

4.22 - MEMOIRES INTERMEDIAIRES D'ENTREE et de SORTIE

Les entrées et sorties sont réalisées avec le tableau de connexion PASO , sur l'unité 533.
Pour cette raison , les zones d'entrée et de sortie utilisées par le Compileur sont chargées de la façon suivante :

4.22.1 - ZONE D'ENTREE 1951 - 1960

1951	←					ADRESSE A				→
1952	←					FACTEUR F				→
1953	←					INSTRUCTION S ^{te} I				→
1954	←					CODE OPERATION	←	INDEX F	←	INDEX I
1955	←					REMARQUES				→
1956	←					REMARQUES				→
1957		0	0	0	0	0	0	X	X	X
1958		0	0	0	0	0	0	X	X	X
1959		0	0	0	0	0	0	X	X	X
1960		0	0	0	0	0	0	0	8/0	8/0

Type (0 si blanc) _____

8 si type non blanc _____

8 si signe négatif _____

}

Partie

alphabétique

}

A

F

I

Partie

absolue

Information de

contrôle

4.22.2 - ZONES DE SORTIE

Le Compileur utilise 10 zones de sortie , chaque zone correspondant à un modèle de carte.

1527 - 36 et 1877 - 86 pour BBL x

1577 - 86 pour une carte K

1627 - 36 ; 1677 - 86 et 1727 - 36 pour les macro-instructions arithmétiques

1777 - 86 ; 1827 - 36 et 1727 - 36 pour les macro-codes en ZZ

1927 - 36 pour la carte Z

1977 - 86 pour la reproduction des cartes PASO standards ou des instructions .

Dans ces zones de perforations , certains Mots invariables sont chargés de façon permanente.

Dans le schéma ci-joint , ces Mots permanents sont hachurés et l'on trouve également sur la dernière ligne , l'entrée dans le Sous-programme PFO correspondant à chaque zone .

4.23 - PLAN D'ENCOMBREMENT DU COMPILEUR SUR LE TAMBOUR

ZONES DE SORTIE

←BBL X → ←KFxx → Carte (A) Mxx (F) (I) → Carte (A) MZZ Δ ₁ Δ ₂ → ←BBL X → ←Carte Z → Carte Standard→										
1527	1577	1627	1677	1727	1777	1827	1877	1927	1977	
	—A—	—A—			—A—	AB99T				—A—
1xxx	—I—	—F—	—A—	—I—	—AB99T—	—Δ ₁ —	—k ₁ —			—F—
1999	Okkk			17xx		—Δ ₂ —	—k ₂ —	omnp		—I—
PRA	EDI	OO	RAD	AAG	RAD	OO	PRA	PDT		COP
BBL	KFx	Mxx	Mxx	Mxx	MZZ	MZZ	BBL	CONST		REMAR
X	x	—F—	—F—	—F—	—Δ ₁ —	—Δ ₁ —	X	ANTE(S)		QUES
N°	N°	N°	N°	N°	N°	N°	N°	N°	N°	N°
1039	1295	1399	1501	1445	1197	1299	1723	1601		1289

CHAPITRE V

Annexes

ANNEXES

Les annexes sont au nombre de sept :

- Annexe 1 : Analyse de l'A.D.P.
- Annexe 2 : Analyse de l'A.S.D.P.
- Annexe 3 : Analyse de l'A.T.P.
- Annexe 4 : Analyse de l'A.Cx.S.P.
- Annexe 5 : Analyse de l'A.Cx.D.P.
- Annexe 6 : Analyse de l'A.Cx.SD.P.
- Annexe 7 : Analyse du Compileur

1- REMARQUES POUR LA LECTURE DES ANNEXES

STRUCTURE D'UNE LIGNE

Une analyse a pour but d'étudier toutes les instructions d'un programme . L'étude d'une seule instruction nécessite l'écriture d'une ligne que nous décomposerons en 4 zones indiquées sur l'exemple ci-dessous :

(33)	1904 21 1715 1868	(19)	(1715)	= - - - - à - - - -
Zone 1	Zone 2	Zone 3	Zone 4	

Zone 1 : Le nombre entre parenthèses placé dans cette zone sert de référence à l'instruction correspondante . Ceci s'avère nécessaire chaque fois que nous rencontrons une instruction à laquelle nous sommes certains d'être renvoyés à un moment quelconque du programme . (Par exemple : première instruction d'une boucle)

Zone 2 : Nous y trouvons l'adresse de l'instruction à étudier et cette instruction en langage machine.

Zone 3 : Elle n'est utilisée que dans le cas où l'instruction à exécuter immédiatement après l'instruction présente , n'est pas étudiée à la ligne suivante . Nous trouvons alors en zone 3 un nombre entre parenthèses que nous recherchons dans la zone 1 d'une autre ligne afin de poursuivre l'étude du programme .

Zone 3 - Cas particuliers des tests : Dans le cas d'un test :

1873	46	1776	1777
		(9)	(10)
		Zone 3	

étant donné que la réponse au test peut être OUI ou NON , les repères de la zone 3 sont placés sous la ligne , au regard de l'adresse correspondant à la réponse au test . Si la réponse au test est :

OUI , la suite doit être prise en 1776 au repère (9) en Zone 1

NON, la suite doit être prise en 1777 au repère (10) en Zone 1

Zone 4 - Elle est réservée aux commentaires . Elle comportera le plus souvent le contenu des mémoires de travail , des accumulateurs ou toutes autres indications nécessaires à la bonne compréhension du programme .

Par exemple : (1715) = ----- à -----

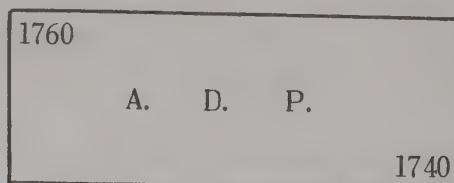
signifie que la mémoire 1715 reçoit la mantisse a .

2 - REPRESENTATION D'UN SOUS-PROGRAMME

Nous avons convenu de schématiser un sous-programme à l'aide d'un rectangle à l'intérieur duquel nous plaçons :

- au centre , la nature du sous-programme ,
- dans le coin supérieur gauche , l'adresse de la mémoire par laquelle se fait l'entrée dans le sous-programme ,
- dans le coin inférieur droit , l'adresse de la mémoire de sortie .

Par exemple :



Sous-programmes utilisés pour les arithmétiques :

1° - A . V F . Arithmétique Virgule Flottante , simple précision

2° - A . D . P . et A . SD . P .

Pour leurs séquences d'entrée , voir paragraphes 2.113 et 2.123 .

ANNEXE 1

ARITHMETIQUE DOUBLE PRECISION

A . D . P . ou B , B , L , A

pour Nombres Réels , Virgule Flottante FLAIR

Double Précision

Entrées dans l'Arithmétique

1757 : Addition	DAA
1758 : Soustraction	DAS
1759 : Addition de valeur absolue	DAV
1760 : Division	DDV
1761 : Multiplication	DMU
1762 : Soustraction de valeur absolue	DSV
1763 : Transfert du P.A. vers la mémoire générale	DTR
1764 : Transfert du nombre dans le P.A.	DZA
1765 : Transfert de l'opposé du nombre dans le P.A.	DZS
1766 : Transfert de la valeur absolue de nombre dans le P.A.	DZV
1767 : Transfert de l'opposé de la valeur absolue du nombre dans le P.A.	DZW
1797 : Chargement de la distance entre Mot 1 et Mot 2	DZZ

Macro-Code Symbolique

Sorties de l'Arithmétique

en 1740 d'une manière générale

en 8003 pour les opérations DTR et DZZ .

Mémoires à contenu variable

Pseudo-Accumulateur	
1700	00 0000 0000
exposant du résultat ou du premier opérande.	
1715	a ₁
Mot 1 du premier opérande ou du résultat de l'opération.	
1730	a ₂
Mot 2 du premier opérande ou du résultat de l'opération.	
Pseudo-Distributeur	
1705	00 0000 0000
exposant du second opérande	
1717	b ₁
Mot 1 du second opérande	
1731	b ₂
Mot 2 du second opérande	
1717 et 1731 sont également utilisés comme mémoires de travail dans le sous-programme de division.	
1711	Mémoire de travail
1732	00 D' 0000+
D' est la distance entre le Mot 1 et le Mot 2 des nombres traités.	
1740	(Inst.Sortie)
Instruction à prendre à la suite de l'opération en cours.	
1746	YY...c ₁
Mot 1 du résultat chargé à la fin du SPS.	
Mémoire de travail également.	
1869	{00 0800 1850 Orientation: 8: transfert vers le PA 00 0900 1850 9: opération arithmétique {35 002-2 1802 Recadrage vers la gauche ou la droite {31 002-2 1802 de la mantisse du résultat. Le nombre des décalages est 2-2 calculé par le SPS.
Mémoire de travail également.	
1704	65 (F') 1846
Appel du Mot 2 dans 8002	
1716	Orientation
on y charge la première instruction du SPO sélectionnée.	
1729	31 oodu 1709
oodu = q-β , différence des exposants et nombre de décalages à droite avec arrondi à effectuer sur l'une des mantisses.	
1748	31 xxxx 1802
Instruction de recadrage de la mantisse du résultat. xxxx est calculé par le SPS.	
1780	31 oodu 1863
oodu = q-β , différence des exposants et nombre de décalages à droite avec arrondi à effectuer sur l'une des mantisses	
1846	10 (F) 1716
appel du Mot 1 dans 8003	
1901	35 xxxx 1802
Instruction de recadrage de la mantisse du résultat. xxxx est calculé par le SPS.	
1947	24 (F') 8003
Transfert du Mot 2 du résultat vers F'	
1992	24 (F) 1997
Transfert du Mot 1 du résultat vers F'	

Mémoires à contenu fixe

1713	00 0900 1850	Orientation vers le sous-programme de division.
1714	00 0800 1850	Orientation vers le sous-programme de multiplication.
1741	01 0333 8000	La distance D' est nulle ou bien n'a pas été chargée en 1732.
1779	01 0222 8000	Le programme aurait à effectuer une division dont le diviseur est nul.
1844	01 0111 8000	Arrêt Overflow: l'exposant du résultat est devenu supérieur à 99.
1744	00 0010 0000 +	
1852	00 0000 0100 +	
1855	00 0008 0000 +	
1861	00 0000 0052 +	
1879	99 9999 9999 +	
1914	00 0000 0048 +	
1938	00 0000 0010 +	
1998	02 0000 0000 +	
1962	00 0000 0001 +	

1° Sous-Programme d'ENTREE

L'entrée dans l'Arithmétique est spécifique de chaque opération; on distingue 6 opérations arithmétiques et 4 transferts vers le PA. A l'entrée, les contenus de 8003 et 8002 sont les suivants:

(8003) = Instruction de Sortie

(8002) = 00 F xxxx, F étant l'adresse du Mot 1 du nombre traité par l'opération actuelle.

1757	69 1720 1726	
1758	69 1721 1726	
1759	69 1722 1726	On charge en 8001 la première instruction du SPO correspondant à l'opération.
1760	69 1713 1726	
1761	69 1714 1726	
1762	69 1723 1726	
1764	69 1817 1728	On charge en 8001 la première instruction du SPO correspondant au transfert.
1765	69 1721 1728	
1766	69 1722 1728	
1767	69 1723 1728	

1726 24 1716 1719
1719 69 1713 1816

(8001) transféré en 1716.
(8001) = 0009001850+
9 en position 7 : opération
arithmétique.

1728 24 1716 1819
1819 69 1714 1816

(8001) transféré en 1716
(8001) = 0008001850+
8 en position 7 : opération de
transfert vers le PA.

1816 24 1869 1822
1822 21 1740 1743
1743 69 1846 1749
1749 22 1846 1799

(1869) = 60 0800 1850
(1740) = Instruction de Sortie
(8002) = 00 F xxxx
(1846) = 10 F 1716
Instruction d'appel du Mot 1
dans 8002.

1799 65 1732 1737
1737 45 1840 1741

(8002) = 00 D' 0000 +
(1741) = 01 0333 8000 +
La distance D' étant nulle ou
n'ayant pas été chargée, il y
arrêt du déroulement du pro-
gramme.

1840 15 1846 1701
1701 69 1704 1707
1707 22 1711 8001

(8002) = 10 (F+D') 1716
(8001) = 65 F' 1846
Instruction d'appel du Mot 2

8001 65 F' 1846
1846 10 F' 1716

(8003/2) = $\beta\beta\dots b_1\dots\dots b_2\dots\dots$
La suite des instructions est
prise en 1716 qui contient la
première instruction du SPO sé-
lectionné.

REMARQUE

Chargement de la distance D' en 1732. Les contenus de
8003 et 8002 étant les suivants :

(8003) = Instruction de Sortie
(8002) = 00 D' 0000 +

l'instruction de chargement est :

1797 20 1732 8003.

(1732) = 00 D' 0000 +

2° Sous-Programmes de TRANSFERT

2.1 Transferts vers le PA

A l'entrée dans les sous-programmes de transfert vers le
PA, 8003 et 8002 contiennent le nombre B sous forme 2 Mots.

(8003/2) = $\beta\beta\dots b_1\dots\dots b_2\dots\dots$

Transfert de B : (1716) = (1817)

(1) 1817 20 1730 1733 (1730) = $\dots b_2\dots\dots$

1733 21 1746 1849 (1746) = $\beta\beta\dots b_1\dots\dots$

1849 30 0008 1867

1867 21 1869 1872 (8003) = (1869) = 00000000 $\beta\beta$

1872 65 8002 1781

1781 30 0002 1837

1837 20 1715 1718 (8002) = (1715) = 00 $\dots b_1\dots\dots$

1717 67 1869 1823

1823 20 1700 1703 (8002) = (1700) = 00000000 $\beta\beta$ +

1703 65 1730 1735

1735 10 1746 1740 (8003/2) = $\beta\beta\dots b_1\dots\dots b_2\dots\dots$

1740 Inst. de Sortie.

Transfert de -B : (1716) = (1721) et (1869) = 0008001850

(2) 1721 21 1746 1899 (1746) = $\dots b_1\dots\dots$

1899 66 8002 1807

1807 11 1746 1720 (3) (8003/2) = $-(\beta\beta\dots b_1\dots\dots b_2\dots\dots)$

(3) 1720 69 1869 1922 (8001) = 00 0800 1850

1922 97 1817 1727
(1) La suite des instructions est
prise en 1817

Transfert de +B : (1716) = (1722) et (1869) = 0008001850

1722 46 1721 1720

(2) (3) Si B est positif, la suite est
en 1720 : Addition

(2) (3) Si B est négatif, la suite est
prise en 1721 : Soustraction

Transfert de -|B| : (1716) = (1723) et (1869) = 0008001850

1723 46 1720 1721

(3) (2) Si B est positif, la suite est
prise en 1721 : Soustraction

(3) (2) Si B est négatif, la suite est
prise en 1720 : Addition.

2.2 Transfert vers la mémoire générale.

Le résultat de l'opération est chargé sous la forme 2 Mots dans les mémoires 1746 et 1730 :

$$(1746) = \gamma\gamma\dots c_1\dots \text{ et } (1730) = \dots e_2\dots$$

Il doit être transféré dans les mémoires d'adresses F et F' ($F' = F + D'$), F étant l'adresse indiquée en position d'adresse facteur de 8002.

A l'entrée :
 $(8003) =$ Instruction de Sortie
 $(8002) = 00 \text{ F } xxxx$

1763 69 1992 1945 $(8002) = 00 \text{ F } xxxx$
 1945 22 1992 1995 $(8001) = (1992) = 24 \text{ F } 1997$
 1995 15 1732 1894 $(8001) = 00 \text{ D' } 0000$
 1894 69 1947 1944 $(8002) = 00 \text{ F' } xxxx$
 1944 22 1947 1994 $(8001) = 24 \text{ F' } 8003$

1994 69 1746 1992 $(8001) = \gamma\gamma\dots c_1\dots$
 1992 24 F 1997
 1997 69 1730 1947 $(8001) = \dots e_2\dots$
 1947 24 F' 8003 $(8003) =$ Instruction de Sortie

3° Sous-Programme d'ADDITION

A l'entrée dans le Sous-Programme, le second opérande B se trouve dans 8003/2 sous la forme :

$$(8003/2) = \beta\beta\dots b_1\dots \dots b_2\dots$$

et puisqu'il s'agit d'une opération arithmétique, 1869 contient un 9 en position 7 :

$$(1869) = 00 \text{ 0900 } 1850$$

3.1 Opération mise ramenant à l'Addition

3.1.1 SOUSTRACTION: $(1716) = (1721)$.

(5) 1721 21 1746 1899 $(1746) = \beta\beta\dots b_1\dots$
 1899 66 8002 1807
 1807 11 1746 1720 (4) $(8003/2) = -(\beta\beta\dots b_1\dots \dots b_2\dots)$

3.1.2 Addition de VALEUR ABSOLUE: $(1716) = (1722)$

1722 46 1721 1720 (5) (4)
 Si B est positif la suite est prise en 1720 : Addition
 Si B est négatif, la suite est prise en 1721 : Soustraction

3.1.3 Soustraction de VALEUR ABSOLUE: $(1716) = (1723)$

1723 46 1720 1721 (4) (5)
 Si B est positif, la suite est prise en 1721: Soustraction
 Si B est négatif, la suite est prise en 1720: Addition

3.2 ADDITION

3.2.1 Chargement de B dans le PP.

Le second opérande B se trouve dans 8003/2 sous la forme 2 Mots après un éventuel changement de signe.

(4) 1720 69 1869 1922 $(8001) = 00 \text{ 0900 } 1850$
 1922 97 1817 1727 La suite des instructions est prise en 1727.
 1727 20 1731 1734 $(8002) = (1731) = \dots b_2\dots$
 1734 30 0008 1803 $(8003) = (1746) = 00000000\beta$
 1803 21 1746 1949
 1949 65 8002 1857
 1857 30 0002 1813 $(8002) = (1717) = 00\dots b_1\dots$
 1813 20 1717 1820
 1820 67 1746 1801
 1801 20 1705 1708 $(8002) = (1705) = 00000000|\beta|+$
 Le second opérande B se trouve donc chargé dans le PD sous la la forme 3 Mots :
 $(1705) = 00000000|\beta|+$
 $(1717) = 00\dots b_1\dots$
 $(1731) = \dots b_2\dots$

3.2.2 Etude des cas particuliers.

1708 45 1712 1863 (6)
 En 1863, le nombre B est nul, il ne reste qu'à charger a avant de passer au sous-pro-

gramme de Recadrage

1712 67 1700 1805
 1805 45 1808 1709 (15)
 En 1709, le nombre A est nul.
 Il faut transférer β en 1700 et inscrire b dans 8003-8002 avant de passer au sous-gramme de Recadrage.

(8002) = 00000000 α +
 (8003) = 0000000000 +

3.23 Détermination de l'exposant γ

1808 18 1705 1809
 1809 45 1812 1913 (8)
 1812 35 0004 1873
 1873 46 1776 1777 (9) (10)
 En 1913 $\alpha - \beta = 0$, donc $\gamma = \alpha = \beta$
 En 1776 $\alpha - \beta < 0$, donc $\gamma = \beta$
 En 1777 $\alpha - \beta > 0$, donc $\gamma = \alpha$

3.24 Calcul de la mantisse α

ler Cas $\gamma = \alpha = \beta$

Il n'y a aucun décalage à effectuer sur l'une ou l'autre des mantisses.

1913 65 1730 1785
 1785 10 1715 1853
 (14) 1853 15 1731 1835
 1835 10 1717 1821
 (8003/2) = 00... a_1 a_2
 (8003/2) = 0?.. a_1 a_2

L'addition des 2 mantisses étant terminée, la suite des instructions est prise en 1821, à l'entrée du Sous-Programme de Recadrage.

2ème Cas $\gamma = \beta$

On commence par fractionner le nombre des décalages en tranches de 10 au maximum, puis après avoir décalé a de "oodu" positions à droite et avec arrondi, on additionne b.

(9) 1776 69 1729 1782
 1782 22 1729 1832
 1832 15 1744 1800
 1800 46 1804 1854 (11)
 (8002) = 00 oodu 0000 < 0
 (8001) = {1729} = 31 oodu 1709
 {1744} = 00 0010 0000 +
 (8002) = 00 00($\alpha - \beta + 10$) 0000

(1855) = 00 0008 0000 +
 (8002) = 00 00($\alpha - \beta + 10$) 0000

1804 15 1855 1909
 1909 46 1912 1814 (12) (13)

(11) 1854 60 1715 1919
 1919 15 1730 1729
 1729 31 oodu 1709
 (15) 1709 69 1705 1858
 1858 24 1700 1853 (14)

(8001) = (1700) = 00000000 β +

(12) 1912 65 8003 1709 (15)
 Nous avons: $\alpha < \beta - 10$
 A est considéré comme nul devant B: on remet 8003/2 à zéro avant de transférer exposant et mantisse de B.

(13) 1814 65 1715 1729
 1729 31 oodu 1709 (15)
 Nous avons: $\beta - 10 \leq \alpha < \beta - 10$
 On néglige a_2 et on effectue un décalage a_2 à droite avec arrondi de "u" positions sur a_1 .
 (8003/2) = 00000000000..0.. a_1 ..

3ème Cas $\gamma = \alpha$

On commence par fractionner le nombre de décalages en tranches de 10 au maximum, puis après avoir décalé b de "oodu" positions à droite avec arrondi, on additionne a.

1777 69 1780 1783
 1783 22 1780 1833
 1833 66 8002 1841
 1841 15 1744 1999
 1999 46 1702 1903 (16)
 1702 15 1855 1859
 1859 46 1862 1963 (17) (18)

(8002) = 00 00($\beta - \alpha + 10$) 0000

(16) 1903 65 1731 1885
 1885 10 1717 1780
 1780 31 oodu 1863
 1863 15 1730 1935
 Nous avons: $\alpha - 10 \leq \beta < \alpha$
 Une seule tranche de "u" décalages à droite avec arrondi sur b
 (8003/2) = 000.0.. b ...

1935 10 1715 1821

(8003/2)=0?.....c₁.....c₂....
L'addition des 2 mantisses est terminée et la suite des instructions est prise en 1821, à l'entrée du Sous-Programme de Recadrage.

(17) 1862 65 8003 1863 (6)

Nous avons $\beta < \alpha - 48$
B est considéré comme nul devant A: on remet 8003/2 à zéro avant d'y transférer a.

(18) 1963 65 1717 1780

Nous avons $\alpha - 48 \leq \beta < \alpha - 40$
On néglige b₂ et on effectue un décalage 2 à droite avec arrondi de "u" positions sur b₁

1780 31 0018 1863 (6) (8003/2) = 000000000000.0. b₁.

Remarque

A la sortie du Sous-Programme d'ADDITION et à l'entrée du Sous-Programme de RACADRAGE, le résultat de l'addition (ou de la soustraction), se trouve distribué comme suit :

exposant brut en 1700 : 000000000
mantisse brute en 8003/2 : 0?.....c₁.....c₂....

4° Sous-Programme de MULTIPLICATION

4.1 Transfert de B dans le PD.
Recadrage de a.

Ces instructions sont communes aux Sous-Programmes de Multiplication et de Division. Les mantisses a et b doivent être chargées dans le PA et le PD sous la forme suivante :

(1715) =a₁..... (1730) =a₂....00
(1717) =b₁..... (1731) =b₂....00

A l'entrée, le second opérande B est chargé dans 8003/2 sous la forme :

(8003/2) = β ...b₁...b₂....

Et l'on a (1716) = 00.0800 1850

1850 21 1705 1908 (8002) = (1705) = β ...b₁...

- 178 -

1908 35 0002 1815

1815 20 1731 1784 (8002) = (1731) =b₂....00

1784 21 1717 1870 (8003) = (1717) =b₁....

1870 65 1730 1736

1736 10 1715 1969 (8003/2) = 00....a₁.....a₂....

1969 35 0002 1725

1725 21 1715 1818 (8003) = (1715) =a₁....

1818 20 1730 1883 (8002) = (1730) =a₂....00

1883 67 1705 1710 (8002) = $|\beta$...b₁...|+

1710 35 0002 1917

1917 21 1705 1810 (8003) = (1705) = 00000000| β |+

4.2 Etude des cas particuliers.

1810 67 1700 1905 (8002) = 00000000| α |+

1905 45 1860 1910 (19)
En 1910, le multiplicande ou le dividende étant nul, le résultat de la multiplication ou de la division est nul et déjà chargé dans le PA puisque A est nul. La suite est prise à la fin du Sous-Programme de Sortie.

1860 67 1705 1811 (8002) = 00000000| β |+

1811 69 1716 1920 (8001) = 00 0800 1850

1920 97 1923 1775 (20) (21)
Séparation des Sous-programmes de Multiplication et de Division.

(20) 1923 45 1826 1827 (22) (23)

En 1827, le multiplicateur est nul, donc le produit également.

(23) 1827 21 1700 1904

(8003) = 00000000000 +

(33) 1904 21 1715 1868

Le PA est chargé de zéros.

1868 21 1730 1910 (19)

4.3 Calcul de l'exposant

(22) 1826 17 1700 1706

(8002) = 00000000(β + α) +

1706 16 1861 1865

(8002) = 00000000(β + α -52)

1865 20 1700 1806

(1700) = 00000000(γ -52)

On charge dans 1700 l'exposant du résultat diminué de 2 unités en raison de la correction d'exposant qui doit avoir lieu dans le Sous-Programme de Recadrage.

4.4 Calcul de la mantisse c_2

4.4.1 Calcul des produits partiels.

Dans ce calcul, on négligera le produit $a_2 \cdot b_2$. Ce fait on introduit une erreur inférieure en lue à 1×10^{-10} alors que l'erreur introduite par le recadrage de la mantisse reste inférieure à $0,5 \times 10^{-10}$ soit 50×10^{-12} ce qui reste négligeable.

(27) 1806 60 1715 1970

1970 19 1717 1887

1887 20 1711 1864

1864 21 1746 1900

1900 60 1715 1871

1871 19 1731 1951

1851 31 0000 1825

1825 20 1869 1972

1972 60 1730 1836

1836 19 1717 1937

1937 31 0000 1911

4.4.2 Calcul de c_1 et c_2

Le calcul de c_3 est inutile puisque $c(a_1 \cdot b_2)$ et $c(a_2 \cdot b_1)$ sont déjà majorés.

1911 15 1869 1973

1973 25 1711 1915

1915 10 1746 1821

(1869) = $c(a_1 \cdot b_2)$

(1711) = $D(a_1 \cdot b_1)$

(1746) = $c(a_1 \cdot b_1)$ et

(8003/2) = $\dots c_1 \dots c_2 \dots$

La mantisse du résultat est chargée en 8003/2 et l'exposant pré-corrigé en 1700 ; la suite des instructions est prise en 1821, à l'entrée du Sous-Programme de Recadrage.

5° Sous-Programme de DIVISION

Le contenu de 1716 est alors :

(1716) = 00 0900 1850

Après le chargement de B, le recadrage de a et l'étude du cas particulier où $a=0$, puisque 1716 contient un 9 en position 7, la suite des instructions est prise en 1775.

5.1 Cas particulier $\beta=0$

(21) 1775 45 1778 1779

(8002) = 00000000 β +

En 1779, un arrêt programmé interrompt le déroulement du programme puisque le diviseur se trouve nul.

(1779) = 01 0222 8000

5.2 Calcul de l'exposant

1778 67 1700 1856

1856 18 1705 1961

1961 15 1914 1921

1921 20 1700 1906

(8002) = 00000000 $|a|$ +

(8002) = 00000000 $(a-\beta)$

(1914) = 0000000048+

(8002) = (1700) = 00000000 $(e-\beta+48)$

On charge dans 1700 l'exposant diminué de 2 unités en raison de la correction d'exposant qui doit avoir lieu dans le Sous-Programme de Recadrage.

5.3 Calcul de M

5.3.1 Cadrage de a dans 8003/2

1906 67 1715 1971

1971 18 1717 1724

1724 46 1877 1828

(24) (25)

(8002) = $|a_1| - |b_1|$

(8002) = 00000000 (γ)

(1962) = 0000000001+

(1700) = 00000000 $(\gamma+1)$

On majore l'exposant du quotient d'une unité pour compenser le décalage à droite que doit subir

la mantisse a pour rendre le calcul de M possible.

1964 65 1730 1936
1936 10 1715 1874
1874 31 0001 1824 (26) $(8003/2) = 0 \dots a_1 \dots a_2 \dots 0$

1877 65 1730 1886
1886 10 1715 1824 (26) $(8003/2) = \dots a_1 \dots a_2 \dots 00$

(26) 1824 14 1717 1927
1927 20 1715 1918
1918 60 8003 1875
1875 64 1717 1878
1878 20 1730 1933

5.4 Calcul du terme correctif L

5.4.1 Existence et calcul de f

1933 65 1731 1987
1987 45 1890 1863 (6)

1890 14 1717 1928
1928 20 1869 1924
1924 60 8003 1831
1831 64 1717 1829
1829 20 1711 1965

1965 10 1869 1974
1974 31 0001 1881
1881 60 8002 1739
1739 19 8003 1866
1866 65 8003 1925
1925 31 0008 1745

 $(8002) = \dots b_2 \dots 00$
En 1863, b_2 est nul. Il est inutile de calculer f. La division est terminée.
 $(1717) = \dots b_1 \dots$
 $(8002) = (1869) = 000000000f_1$
 $(8003) = \text{reste}$
 $(8002) = (1711) = \dots f_2 \dots$

 $(8003/2) = 000000000f_1 \dots f_2 \dots$
 $(8003/2) = 0000000000 \dots f_1 \dots$
 $(8003) = \dots f_1 \dots$
 $(8003/2) = \dots f_2 \dots$

 $(8002) = 0000000.f_2^2$
On ne conserve de f_2^2 que les chiffres de rang -18, -19 et -20.

5.4.2 Calcul de L = $1-f+f^2$

1745 16 1711 1916
1916 11 1869 1975

1975 15 1879 1834
1834 10 8001 1843
1843 15 1962 1968

1968 20 1731 1884
1884 21 1717 1806 (27)

Les 2 tranches du coefficient L sont chargées en 1717 et 1730 en vue de l'utilisation du Sous-Programme de Multiplication pour le calcul du produit MxL.

5.5 Calcul du produit MxL

Les termes M et L ayant été chargés dans les mémoires 1715, 1730 et 1717, 1731, on peut utiliser le Sous-Programme de Multiplication.

(27) 1806
Sous-Programme de Multiplication
1821

$(8003/2) = \dots c_1 \dots c_2 \dots$
La division étant terminée, la suite des instructions est prise en 1821 à l'entrée du Sous-Programme de Recadrage.

6° Sous-Programme de RECADRAGE

A l'entrée dans le Sous-Programme de RECADRAGE, l'exposant brut du résultat est chargé en 1700 et la mantisse brute se trouve en 8003/2.

Le but du Sous-Programme est de recadrer la mantisse sur la position 8 de 8003 afin de libérer les positions 10 et 9 dans lesquelles l'exposant corrigé du résultat devra être chargé.

(32) 1821 21 1715 1876
1876 20 1730 1934

 $(8003) = (1715) = \dots c_1 \dots$
 $(8002) = (1730) = \dots c_2 \dots$

1934 45 1738 1827

(23)

En 1827, la mantisse résultat étant nulle, le résultat l'est également. Il faut charger 0 en 1700. Voir en (23) au SP de Multiplication.

1738 44 1891 1742

(28)

1891 36 0000 1966

(8002) = xxxxxxx02

Z = nombre de décalages à gauche

1966 35 0008 1838

1838 67 8002 1747

1747 16 1998 1926

1926 30 0004 1888

(8002) = 02 0000 0000 +

(8002) = 0(Z-2) 0000 0000

(8002) = 00 000(Z-2) 0000

En 1893, Z=2 : aucun recadrage de la mantisse n'est nécessaire.

1888 45 1842 1893

(29)

1842 46 1845 1896

1845 69 1748 1976

(30)

(8001) = 31 0000 1802

Nous avons $Z < 2$, il faut donc décaler la mantisse vers la droite pour libérer les positions 10 et 9 de 8003

1896 69 1901 1976

(30)

(8001) = 35 0000 1802

Nous avons $Z > 2$, il faut donc décaler la mantisse vers la gauche pour la recadrer sur la position 8 de 8003.

(30) 1976 22 1869 1929

1929 30 0004 1839

1839 66 8002 1847

1847 15 1700 1830

(8001) = $\frac{1}{5}$ 000 [Z-2] .1802

(8002) = 00 0000 000(Z-2)

(8002) = 00 0000 000(2-Z)

(8002) = (1700) = 00000000(Y+2-Z)

1880 60 1715 1930

1930 15 1730 1869

1869 $\frac{1}{5}$ [Z-2] 1802

(8003/2) =c₁.....c₂.....

(8003/2) = 00.....c₁.....c₂.....

La mantisse recadrée est en 8003/2 et la suite est prise en 1802 à l'entrée du Sous-Programme de Sortie.

(28) 1742 67 1700 1931

1931 16 1938 1943

1943 20 1700 1882

(1938) = 00000000010+

(8002) = (1700) = 00000000(Y-10)

Puisque c₁ est nul, on diminue Y de 10 unités et on reprend la suite des recadrages sur c₂

1882 60 1730 1821

(32)

(8003/2) =c₂.....0000000000

7° Sous-Programme de SORTIE

7.1 Test sur la position 9 de 8003

A l'entrée, la mantisse recadrée se trouve chargée en 8003/2, la position 10 de 8003 étant libre, mais la position 9 pouvant ne pas l'être.

(31) 1802 21 1715 1932

1932 20 1730 1988

1988 36 0000 1889

1889 67 8002 1897

1897 16 1862 1939

1939 45 1893 1863

(29) (6)

(8003) = (1715) = 0?.....c₁....

(8002) = (1730) =c₂.....

(8002) = 000000000Q Q=1, ou 2

(8002) = 000000000(Q-1)

En 1863, puisque Q=1, la position 9 était occupée, il faut donc reprendre le Sous-Programme de Recadrage.
Q≠1, donc les positions 10 et 9 sont libres.

7.2 Tests sur l'exposant

L'exposant doit rester un nombre positif de 2 chiffres :

00 ≤ Y ≤ 99

(29) 1893 65 1700 1989

1989 45 1892 1904

(33)

1892 46 1827 1946

(23)

(8002) = 00000000Y±

En 1904, Y est nul. La suite est prise en 1904 pour charger 1715 et 1730 de zéros

En 1827, Y est négatif. On considérera le résultat comme nul. La suite est prise en 1827 pour charger le PA de zéros.

(1852) = 0000000100+

(8002) = 00000000(Y-100)

En 1844, Y ≥ 100. Il y a donc overflow et on trouve l'arrêt 01 0111 8000

7.3 Résultat sous forme "2 Mots"

1993 15 8001 1902

1902 35 0008 1990

1990 20 1711 1941

1941 60 1715 1991

1991 15 1730 1942

1942 46 1895 1996

1895 11 1711 1910

(19)

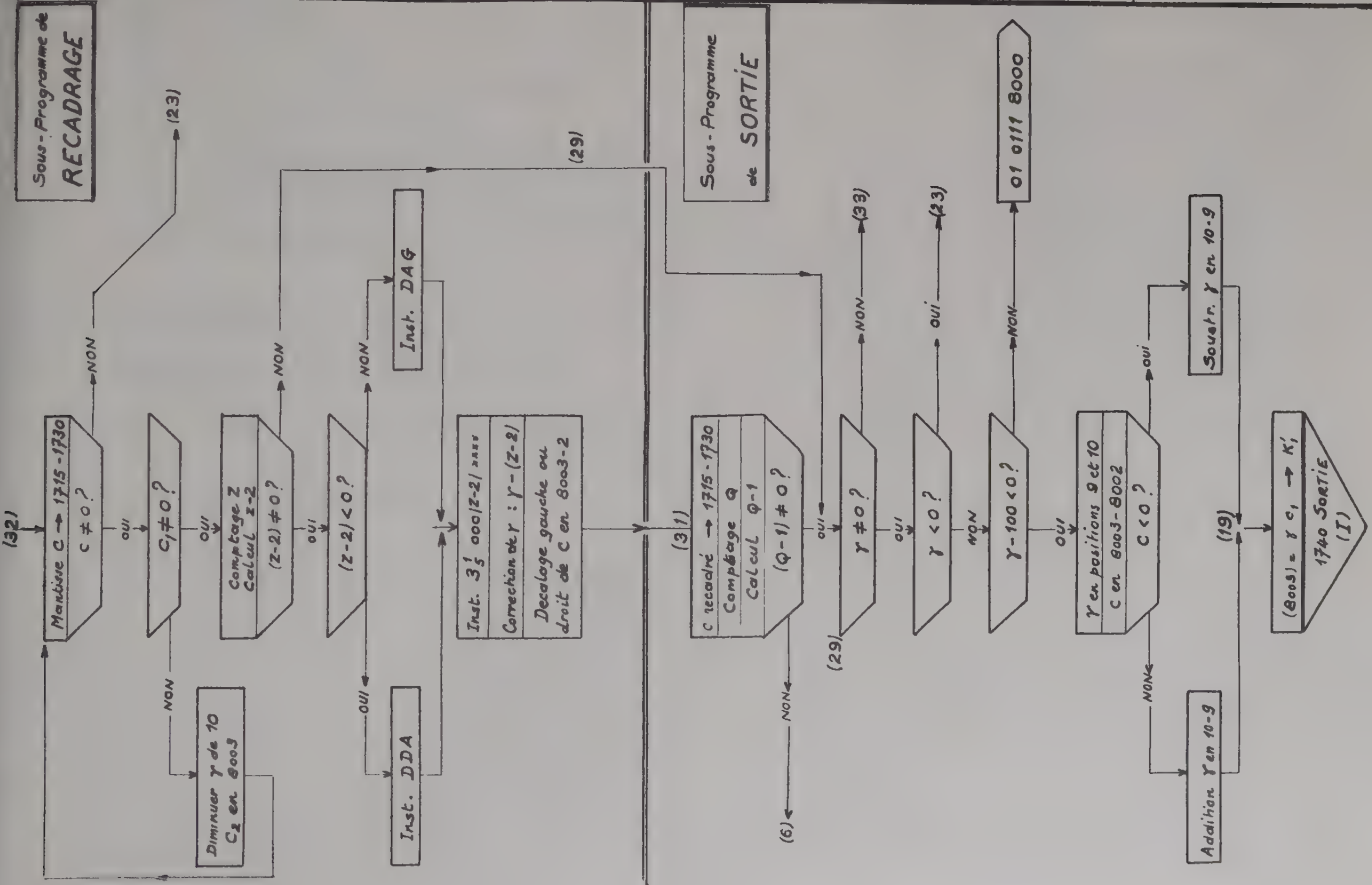
(8002) = Y 000000000+

(1711) = Y 000000000+

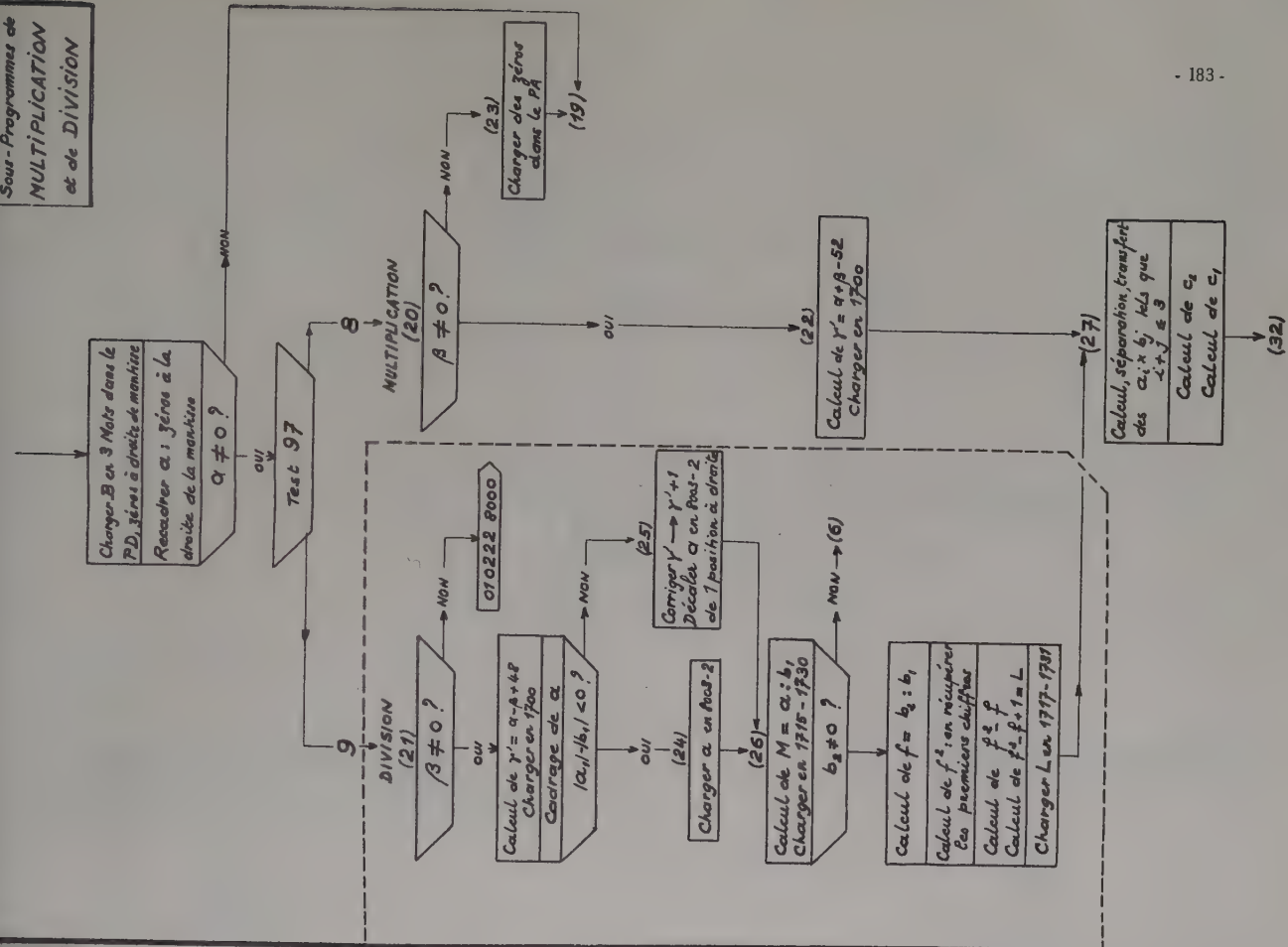
(8003/2) = 00.....c₁.....c₂.....

(8003/2) = YY.....c₁.....c₂..... -

Sous-Programme de
RECADRAGE



Sous-Programmes de
MULTIPLICATION
et de **DIVISION**



ANNEXE 2

ARITHMETIQUE MIXTE , SIMPLE et DOUBLE PRECISION A. SD. P. ou B.B.L. B pour Nombres Réels, Virgule Flottante FLAIR Simple ou Double Précision

Entrées dans l'Arithmétique pour opérations en Simple Précision

Macro-Code Symbolique

1786 : Addition	VAA
1787 : Soustraction	VAS
1788 : Addition de valeur absolue	VAV
1789 : Division	VDV
1790 : Multiplication	VMU
1791 : Soustraction de valeur absolue	VSV
1792 : Transfert dans le P. A.	VZA
1793 : Transfert dans le P.A. de l'opposé	VZS
1794 : Transfert dans le P.A. de la valeur absolue	VZV
1795 : Transfert dans le P.A. de l'opposé de la valeur absolue	VZW

Entrées dans l'Arithmétique pour opérations en Double Précision

1757 à 1767 et 1797 , voir Annexe 1

Sorties de l'Arithmétique

1740 dans tous les cas , sauf
en 8003 pour les opérations DTR et DZZ

Mémoires à contenu variable

Pseudo-Accumulateur.

1700 00 0000 0000 Exposit du premier opérande ou du résultat de l'opération.
1715 .. .a₁..... Mot 1 du premier opérande ou du résultat de l'opération.
1730 .. .a₂..... Mot 2 du premier opérande ou du résultat de l'opération en Double Précision.
En Simple Précision, 1730 est chargé de zéros.

Pseudo-Distributeur

1705 00 0000 0000 Exposit du second opérande
1717 .. .b₁..... Mot 1 du second opérande
1731 .. .b₂..... Mot 2 du second opérande en Double Précision.
En Simple Précision, 1731 est chargé de zéros.

Instructions à adresse facteur variable.

1660 65 F' 1653 Appel du Mot 2 du nombre traité
1688 31 oodu 1829 Ordre de décalage d'une mantisse par rapport à l'autre. La quantité "oodu" est donnée par :
$$oodu = |a - b|$$

1696 10 F' 1703 Appel du Mot 1 du nombre traité.
1779 31 oodu 1659 Ordre de décalage d'une mantisse par rapport à l'autre. Voir ci-dessus.
1780 31 oodu 1781 Ordre de décalage d'une mantisse par rapport à l'autre. Voir ci-dessus.
1802 24 F' 8003 Transfert du Mot 2 du résultat en F'
1816 31 12-21 1900 Ordre de décalage pour normalisation de la mantisse du résultat. La quantité Z-2 est calculée par le SP de RECADRAGE.
1846 60 F' 1803 Ordre d'appel du nombre traité en Simple Précision.
1879 31 oodu 1809 Ordre de décalage d'une mantisse par rapport à l'autre. Voir ci-dessus.
1897 24 F' 1852 Transfert du Mot 1 du résultat en F'.
1964 35 12-21 1900 Ordre de décalage pour normalisation de la mantisse du résultat. Voir ci-dessus.

Mémoires de travail.

Ce sont principalement les mémoires :
1653, 1681 et 1746.

Ces mémoires sont utilisées pour le transfert provisoire d'instructions construites par les différents Sous-Programmes, le transfert provisoire de résultats intermédiaires et comme mémoires d'orientation.

En outre :

1719 00 0800 1650 Séparation de la Simple Précision de la Double Précision.
8 en position 7 : Simple Précision
9 en position 7 : Double Précision.
1732 00 D' 0000 Distance entre les 2 Mots des nombres traités en Double Précision.
1740 Inst.Sortie Instruction de Sortie de l'Arithmétique, c'est à dire Instruction à prendre après l'opération en cours.
1746 YY.c₁..... En fin d'opération, contient le Mot 1 du résultat exprimé sous sa forme "2 Mots"

Mémoires à contenu fixe

1663 00 0900 1650 Le 9 en position 7 signifie : Division ou opération en Double Précision suivant le cas
1664 00 0800 1650 Le 8 signifie : Multiplication ou opération en Simple Précision.
1691 01 0333 8000 La constante D' n'a pas été chargée en 1732 ou bien a été choisie nulle.
1697 01 0222 8000 Le diviseur de la division est nul
1944 01 0111 8000 Arrêt overflow: l'exposant du résultat est devenu supérieur à 99.

1679 00 0008 0000 +
1711 00 0000 0052 +
1714 00 0000 0048 +
1814 00 0000 0001 +
1818 00 0010 0000 +
1914 02 0000 0000 +
1965 99 9999 9999 +
1976 00 0000 0010 +
1844 00 0000 0100 +

1° Sous-Programmes d'ENTRÉE

L'entrée dans l'Arithmétique est spécifique de chaque opération. En Double Précision on distingue 6 opérations arithmétiques, 4 transferts vers le PA et 1 transfert vers la mémoire générale. En Simple Précision, on distingue également 6 opérations, 4 transferts vers le PA. Par contre il n'y a pas de Sous-Programme correspondant à un transfert du résultat en Simple Précision vers la mémoire générale puisqu'il suffit d'un code TRG.

A l'entrée, les contenus de 8003 et 8002 sont les suivants :

(8003) = Instruction de Sortie

(8002) = 00 F xxxx

F étant l'adresse du Mot 1 du nombre traité en Double Précision ou bien l'adresse du nombre traité en Simple Précision.

1.1 Opérations en Double Précision

1757 69 1720 1728

1758 69 1721 1728

1759 69 1722 1728

1760 69 1663 1728

1761 69 1664 1728

1762 69 1723 1728

On charge 8001 de la première instruction du SP Opérationnel correspondant à l'entrée, donc à l'opération.

1764 69 1667 1728

1765 69 1668 1728

1766 69 1669 1728

1767 69 1670 1728

On charge 8001 de la première instruction du SP Opérationnel correspondant à l'entrée, donc à l'opération.

1728 24 1681 1684

1684 21 1740 1693

1693 69 1696 1699

1699 22 1653 1656

1656 65 1732 1687

1687 45 1690 1691

(8001) est transféré en 1681.

(1740) = Inst.Sortie

(8001) = 10 xxxx 1703

(1653) = 10 F 1703

(8002) = 00 D' 0000

En 1691, la distance D' n'ayant pas été chargée, on trouve l'arrêt

01 0333 8000

(8002) = 10 (F+D') 1703

(8001) = 65 xxxx 1653

1690 15 1653 1657

1657 69 1660 1713

1713 22 1746 8001

8001 65 F' 1653

1653 10 F 1703

1703 69 1663 1666

1666 24 1719 1681

(8001) = 65 F' 1653

(8002) =b₂....

(8003) = b₁....

(8001) = 00 0900 1650

(1719) = 00 0900 1650

En 1681 on retrouve la première instruction du SP Opérationnel sélectionné à l'entrée.

Le second opérande B se trouve chargé en 8003-2 et 1719 possède un 9 en position 7 signifiant que l'opération doit être faite en Double Précision.

1.2 Opérations en Simple Précision.

1786 69 1720 1729

1787 69 1721 1729

1788 69 1722 1729

1789 69 1663 1729

1790 69 1664 1729

1791 69 1723 1729

1792 69 1667 1729

1793 69 1668 1729

1794 69 1669 1729

1795 69 1670 1729

On charge 8001 de la première instruction du SP Opérationnel correspondant à l'opération Double Précision, la séparation entre Simple et Double Précision, se faisant à l'intérieur de ces SP Opérationnels par test sur le contenu de 1719.

1729 24 1681 1734

1734 21 1740 1743

1743 69 1846 1749

1749 22 1653 8001

8001 60 F 1803

1803 20 1730 1683

1683 69 1664 1666

1666 24 1719 1681

(8001) est transféré en 1681

(1740) = Inst.Sortie

(8001) = 60 xxxx 1803

(8001) = 60 F 1803

(8003) = aa....a....

(1730) = 0000000000 ±

(1719) = 00 0800 1650

En 1681 on retrouve la première instruction du SP Opérationnel sélectionné à l'entrée.

Le second opérande, formé d'un seul Mot est chargé en 8003 et 1719 possède un 8 en position 7 signifiant que l'opération doit être faite en Simple Précision.

REMARQUE:

Chargement de la distance D' en 1732. Les contenus de 8003 et 8002 étant les suivants :

(8003) = Instruction de Sortie.

(8002) = 00 D' 0000 +

l'instruction de chargement est :

1797 20 1732 8003 (1732) = 00 D' 0000 +

2.1 Transferts vers le PA.

Ce Sous-Programme est commun à la Simple Précision et la Double Précision. Dans le cas de son utilisation par la Simple Précision, le Mot 2 est remplacé par des zéros.

A l'entrée dans les Sous-Programmes de Transfert vers le PA, 8003-2 contient l'opérande B sous la forme 2 Mots :

(8003-2) = $\beta\beta...b_1...b_2...b_2...$

ou (8003-2) = $\beta\beta...b_1...0000000000$

2.1.1 Transfert de B: (1681)=(1667)

(1) 1667 20 1730 1733 (1730) = b_2 ou 00.0
 1733 21 1746 1799 (1746) = $\beta\beta...b_1...$
 1799 30 0008 1817
 1817 21 1653 1706 (1653)=(8003) = 00000000 $\beta\beta$.
 1706 65 8002 1665
 1665 30 0002 1671
 1671 20 1715 1718 (1715)=(8002) = 00... $b_1...$
 1718 67 1653 1707
 1707 20 1700 1853 (1700)=(8002) = 00000000 $\beta\beta$.
 1853 65 1730 1685 (8002) = $..b_2...00.0$
 1685 10 1746 1740 (8003) = $\beta\beta...b_1...$
 1740 Inst.Sortie

L'opérande B ayant été chargé dans le PA, apparaît également sous forme 2 Mots en 8003-2. La suite est prise en 1740 où l'on retrouve l'instruction de Sortie de l'Arithmétique.

2.1.2 Transfert de -B: (1681)=(1668)

On change le signe de B.

(2) 1668 21 1746 1849 (8002) = - ($...b_2...b_2...$)
 1849 66 8002 1807 (8003) = - ($\beta\beta...b_1...$)
 1807 11 1746 1667 (1) La suite est prise en 1667

2.1.3 Transfert de +B: (1681)=(1669)

1669 46 1668 1667 (2) (1) Si B est positif, la suite est en 1667 : Addition
 Si B est négatif, la suite est en 1668 : Soustraction.

2.1.4 Transfert de -B: (1681)=(1670)

1670 46 1667 1668 (1) (2) Si B est positif, la suite est en 1668 : Soustraction
 Si B est négatif, la suite est en 1667 : Addition.

2.2 Transfert vers la mémoire générale

Cette opération n'intéresse que la Double Précision. Le résultat de l'opération arithmétique est chargé sous la forme "2 Mots" dans les mémoires 1746 et 1730 :

(1746) = $\gamma\gamma...e_1...$

(1730) = $...e_2...$

et il doit être transféré dans les mémoires d'adresse F et F', F étant l'adresse indiquée en position d'adresse facteur de 8002.

(8003) = Instruction de Sortie

(8002) = 00 F xxxx

1763 69 1897 1702 (8001) = 24 xxxx 1852
 1702 22 1653 1947 (1653) = 24 F 1852
 1947 15 1732 1997 (8002) = 00 F+D' xxxx
 1997 69 1802 1748 (8001) = 24 xxxx 8003
 1748 22 1681 1848 (1681) = 24 F' 8003
 1848 69 1746 1653

Transfert du Mot 1 du résultat en F.

1852 69 1730 1681

1681 24 F' 8003

Transfert du Mot 2 du résultat en F'.

8003 Inst.Sortie.

Le résultat de l'opération ayant été transféré en F et F', on retrouve en 8003 l'instruction de Sortie.

(15) 1837 65 1715 1653

Nous avons: $\beta - 48 \leq \alpha < \beta - 40$
 On néglige a_2 et on effectue
 une tranche de "u" décalé-
 ges sur a_1 .

1653 31 00du 1659 (8) (8003-2)=00000000000000... a_1 .

3ème Cas: $\gamma = \alpha$.

On commence par fractionner le nombre des décalages en tranches de 10 au maximum, puis après avoir décalé la man-
 tisse b, a droite et avec/arrondi, on lui additionne la
 mantisse b.

(11) 1727 69 1780 1783

(8002) = 00 00($\alpha - \beta$) 0000
 avec "oodu" = $|\alpha - \beta|$
 (8001) = 31 xxx 1781
 (1653) = 31 00du 1781
 (8002) = 00 00($\beta - \alpha$) 0000
 (8002) = 00 00($\beta - \alpha + 10$) 0000

1783 22 1653 1806
 1806 66 8002 1815
 1815 15 1818 1823
 1823 46 1726 1777 (16)
 1726 15 1679 1833
 1833 46 1686 1737 (17) (18)

(16) 1777 65 1731 1835

Nous avons: $\alpha - 40 \leq \beta < \alpha$
 Une seule tranche de "u" dé-
 cales sur b.
 (8003-2) = b décalé.
 Addition de la mantisse a.
 (8003-2) = mantisse résultat.

1835 10 1717 1653
 1653 31 00du 1781
 1781 15 1730 1885
 1885 10 1715 1821 (12)

(17) 1686 65 8003 1781 (7)

Nous avons: $\beta < \alpha - 48$
 B est considéré comme nul
 devant A: On remet 8003-2 à
 zéro avant d'y charger a.

(18) 1737 65 1717 1653

Nous avons: $\alpha - 48 \leq \beta < \alpha - 40$
 On néglige b_2 et on effectue
 une tranche de "u" décala-
 ges sur b_1 .

1653 31 00du 1781 (7) (8003-2)=00000000000000... b_1 .

Il n'y a aucun décalage à effectuer sur l'une ou l'autre
 des mantisses.

(9) 1863 65 1730 1735

1735 10 1715 1654
 1654 15 1731 1785
 1785 10 1717 1821 (12)

(8003-2) = 00... b_1 b_2 ...
 (8003-2)=0?.... c_1 c_2
 Les 2 mantisses ayant été
 additionnées, la suite est
 prise en 1821 à l'entrée du
 SP de RECADRAGE.

2ème Cas: $\gamma = \beta$.

On commence par fractionner le nombre des décalages en tranches de 10 au maximum, puis après avoir décalé la man-
 tisse a, a droite et avec arrondi, on lui additionne la
 mantisse b.

(10) 1676 69 1779 1682

(8002) = 00 00($\alpha - \beta$) 0000
 avec "oodu" = $|\alpha - \beta|$
 (8001) = 31 xxx 1659
 (1653) = 31 00du 1659
 (8002) = 00 00($\alpha - \beta + 10$) 0000

1682 22 1653 1856
 1856 15 1818 1873
 1873 46 1776 1827 (13)
 1776 15 1679 1883
 1883 46 1736 1837 (14) (15)

(13) 1827 60 1715 1819

Nous avons: $\beta - 48 \leq \alpha < \beta$
 Une seule tranche de "u" dé-
 calages sur a.
 (8003-2) = a décalé.

1819 15 1730 1653
 1653 31 00du 1659
 1659 69 1705 1808
 1808 24 1700 1654
 1654 15 1731 1785
 1785 10 1717 1821 (12)

(8) 1659 69 1705 1808

(1700) = 00000000 β +
 Addition de la mantisse b.
 (8003-2) = mantisse résultat.

1808 24 1700 1654
 1654 15 1731 1785
 1785 10 1717 1821 (12)

= 0?.... c_1 c_2
 La suite est prise en 1821
 pour le RECADRAGE de la man-
 tisse.

(14) 1736 65 8003 1659 (8)

Nous avons: $\alpha < \beta - 48$
 A est considéré comme nul
 devant B: on remet 8003-2
 à zéro avant de transférer
 mantisse et exposant de B.

3.4 ADDITION en Simple Précision.

3.4.1 Etude des cas particuliers.

- (5) 1675 45 1678 1829 (19)
 (8002) = 00000000 β .+
 En 1829, le nombre B est nul.
 Il ne reste donc plus qu'à
 charger a_1 dans 8002 avant
 de passer au SP de RECADRAGE.
 (8002) = 00000000 α .+
 En 1809, le nombre A est nul.
 Il faut transférer β en 1700
 et inscrire b_1 en 8002 avant
 de passer au SP de RECADRAGE.
- 1678 67 1700 1805
 1805 45 1858 1809 (20)

3.4.2 Détermination de l'exposant γ

- 1858 18 1705 1859 (8002) = 00000000 ($\alpha - \beta$)
 1859 45 1712 1913 (21)
 1712 35 0004 1923 (8002) = 00 00 ($\alpha - \beta$) 0000
 En 1913, $\alpha - \beta = 0$, donc $\gamma = \alpha = \beta$
 1923 46 1826 1877 (22) (23)
 En 1826, $\alpha - \beta < 0$, donc $\gamma = \beta$
 En 1877, $\alpha - \beta > 0$, donc $\gamma = \alpha$

3.4.3 Calcul de la mantisse α

- 1^{er} Cas: $\gamma = \alpha = \beta$.
 Il n'y a aucun décalage à effectuer sur l'une ou l'autre
 des mantisses.
- (21) 1913 65 1715 1704 (8002) = 00... a_1 ...
 1704 15 1717 1871 (24) Addition de b_1 . En 8002 se
 trouve la mantisse résultat.
 (8002) = 07... a_1 ...
 La suite est prise en 1871
 pour le RECADRAGE de la man-
 tisse.

2^{ème} Cas: $\gamma = \beta$.

- (22) 1826 69 1879 1782 (8002) = 00 00 ($\alpha - \beta$) 0000
 (8001) = 31 xxxx 1809
 1782 22 1681 1884 (1681) = 31 oodu 1809
 avec "odu" = $|\alpha - \beta|$
 (8002) = 00 00 ($\alpha - \beta + 8$) 0000
 1884 15 1679 1934
 1934 46 1937 1738 (25) (26)

- (25) 1937 65 8003 1809 (20)
 Nous avons: $\alpha < \beta - 8$
 A est considéré comme nul
 devant B. On remet 8002 à
 zéro avant d'y charger b_1 .
- (26) 1738 65 1715 1681
 Nous avons: $\beta - 8 \leq \alpha < \beta$
 On décale a_1 de "odu" posi-
 tions à droite avec arrondi.
 (8002) = a_1 décalé.
 (20) 1681 31 oodu 1809 (20)
 1809 69 1705 1908
 1908 24 1700 1704 (1700) = 00000000 β .+
 1704 15 1717 1871 (24) Addition de b_1 .
 (8002) = 07... a_1 ... La suite
 est prise en 1871 pour le
 RECADRAGE de la mantisse ré-
 sultat.

3^{ème} Cas: $\gamma = \alpha$.

- (23) 1877 66 8002 1935 (8002) = 00 00 ($\beta - \alpha$) 0000 -
 1935 69 1688 1741 (8001) = 31 xxxx 1829
 1741 22 1681 1834 (1681) = 31 oodu 1829
 1834 15 1679 1933 (8002) = 00 00 ($\beta - \alpha + 8$) 0000
 1933 46 1836 1887 (27) (28)
 (27) 1836 65 1715 1871 (24)
 Nous avons: $\beta < \alpha - 8$
 B est considéré comme nul de-
 vant A. On charge a_1 en 8002
 avant de passer au RECADRAGE
 en 1871.
- (28) 1887 65 1717 1681
 Nous avons: $\alpha - 8 \leq \beta < \alpha$
 On décale b_1 de "odu" posi-
 tions à droite avec
 arrondi.
 (8002) = b_1 décalé.
 (19) 1681 31 oodu 1829 (24) Addition de a_1 .
 1829 15 1715 1871 (8002) = 07... a_1 ... La suite
 est prise en 1871 pour le
 RECADRAGE de la mantisse ré-
 sultat.

A la Sortie du Sous-Programme d'ADDITION, le résultat
 se trouve distribué comme suit :

Exposant brut en 1700 : 00000000 γ .
 Mantisse brute en 8003-2 : 07... a_1 ... a_2
 Et en Simple Précision (8002) = 07... a_2

4° Sous-Programme de MULTIPLICATION

Les instructions de transfert de l'opérande B dans le PD, du recadrage de la mantisse a, l'étude des cas particuliers et le calcul de l'exposant du produit sont communes à la Multiplication en Simple Précision et à la Multiplication en Double Précision. La séparation ne se fait qu'au niveau du calcul de la mantisse du produit.

4.1 Transfert de B dans le PD Recadrage de a.

Ces instructions sont également communes aux Sous-Programmes de Multiplication et de Division. Les mantisses a et b doivent être chargées dans le PA et le PD sous la forme :

$$\begin{aligned}(1715) &= \dots a_1 \dots & (1730) &= \dots a_2 \dots 00 \\ (1717) &= \dots b_1 \dots & (1731) &= \dots b_2 \dots 00\end{aligned}$$

A l'entrée dans le Sous-Programme, le second opérande B est chargé dans 8003-2 sous la forme :

$$\begin{aligned}(8003-2) &= \beta \beta \dots b_1 \dots \dots b_2 \dots & \text{Double Précision} \\ \text{ou } (8003-2) &= \beta \beta \dots b_1 \dots \dots 0000000000 & \text{Simple Précision.}\end{aligned}$$

Et l'on a $(1681) = 00\ 0800\ 1650$

et $(1719) = 00\ 0800\ 1650$ en Simple Précision
 $(1719) = 00\ 0900\ 1650$ en Double Précision.

$$\begin{aligned}1681\ 00\ 0800\ 1650 \\ 1650\ 21\ 1705\ 1909 \\ 1909\ 35\ 0002\ 1865 \\ 1865\ 20\ 1731\ 1886 \\ 1886\ 21\ 1717\ 1870 \\ 1870\ 65\ 1730\ 1936 \\ 1936\ 10\ 1715\ 1869 \\ 1869\ 35\ 0002\ 1725 \\ 1725\ 21\ 1715\ 1868 \\ 1868\ 20\ 1730\ 1987 \\ 1987\ 67\ 1705\ 1710 \\ 1710\ 35\ 0002\ 1867 \\ 1867\ 21\ 1705\ 1810\end{aligned}$$

$$\begin{aligned}(1705) &= (8003) = \beta \beta \dots b_1 \dots \\ (1731) &= \dots b_2 \dots 00 \\ (1717) &= \dots b_1 \dots \\ (8003-2) &= 00 \dots a_1 \dots \dots a_2 \dots \\ (8003-2) &= \dots a_1 \dots \dots a_2 \dots 00 \\ (1715) &= \dots a_1 \dots \\ (1730) &= \dots a_2 \dots 00 \\ (8003) &= 0000000000 \beta + \\ (1705) &= 000000000 \beta +\end{aligned}$$

4.2 Etude des cas particuliers

1810 67 1700 1855
1855 45 1860 1910
(29)

$$(8002) = 00000000 \beta +$$

En 1910 le multiplicande ou le dividende étant nul, le résultat de la multiplication ou de la division l'est également. et se trouve déjà chargé dans le PA puisque A est nul. La suite est prise en 1910 à la fin du SP de SORTIR.

1860 67 1705 1661
1661 69 1681 1838
1838 97 1841 1843
(30)

$$(8002) = 00000000 \beta +$$

$$(8001) = 00\ 0800\ 1650$$

Séparation des Sous-Programmes de Multiplication et de Division: (30).

1841 45 1694 1695
(31) (37)

En 1695, le multiplicateur est nul, donc le produit également.

(37) 1695 21 1700 1804
(43) 1804 21 1715 1918
1918 21 1730 1910 (29)

$$(1700) = (8003) = 00000000000 +$$

$$(1715) = 0000000000 +$$

$$(1730) = 0000000000 +$$

4.3 Calcul de l'exposant Y

(31) 1694 17 1700 1910
1910 16 1711 1915
1915 20 1700 1854

$$(8002) = 00000000(\beta + d)$$

$$(8002) = 00000000(\beta + d - 52)$$

$$(1700) = 00000000 \beta.$$

Dans 1700, on charge l'exposant du résultat diminué de 2 unités pour tenir compte de la majoration qui aura lieu au cours du SP de RECADRAGE.

4.4 Calcul de la mantisse en Simple Précision

On a alors: $(1719) = 00\ 0800\ 1650$

(32) 1854 60 1715 1919
1919 19 1717 1888
1888 69 1719 1822
1822 97 1775 1927
(33)

$$(8003-2) = a_1 \times b_1$$

$$(8001) = 00\ 0800\ 1650$$

En 1775, la multiplication en Simple Précision, la suite est prise au SP de RECADRAGE.

4.5 Calcul de la mantisse en Double Précision

4.51 Calcul des produits partiels.

Dans ce calcul on négligera le produit $a_i \times b_j$. Ce faisant on introduit une erreur inférieure en valeur absolue à 1×10^{-40} alors que l'erreur introduite par le recadrage de M

(34) 1674 65 1730 1839
1839 10 1715 1971 (35) (8003-2) =a₁.....a₂.....00

5.4 Calcul du quotient en Simple Précision.

On a alors: (1719) = 00 0800 1650

(35) 1971 14 1717 1778 (8002) =c₁.....
1778 69 1719 1972 (8001) = 00 0800 1650
1972 97 1871 1828 En 1871 la division en Simple
(24) Précision est terminée; la
suite est prise en 1871 pour
le RECADRAGE.

5.5 Calcul de M pour la division en Double Précision.

On a alors: (1719) = 00 0900 1650

(35) 1971 14 1717 1778 (8002) =m₁.....
1778 69 1719 1972 (8001) = 00 0900 1650
1972 97 1871 1828

1828 20 1715 1968 (1715) =m₁.....
1968 60 8003 1925 (8003) = reste
1925 64 1717 1878
1878 20 1730 1939 (1730) = (8002) =m₂.....
Les 2 tranches de M sont char-
gées en 1715 et 1730 en vue
de leur utilisation par le SP
de Multiplication pour le cal-
cul de MxL.

5.6 Calcul du terme correctif L pour la Division en Double Précision.

5.6.1 Existence et calcul de f.

1939 65 1731 1989 (8002) =b₂.....00
1989 45 1692 1781 En 1781, b₂ est nul. Il est
(7) donc inutile de calculer L
et le quotient de la division
est M.

1692 14 1717 1928
1928 20 1653 1862 (1653) = (8002) = 000000000f₁
1862 60 8003 1973 (8003) = reste
1973 64 1717 1929
1929 20 1681 1840 (1681) = (8002) =f₂.....

5.6.2 Calcul de L = l - f + f²

1840 10 1653 1912 (8003-2) = 0000000000f₁.....f₂.....
1912 31 0001 1724 (8003-2) = 0000000000.....f₂.....
1724 60 8002 1890 (8003) =f₂.....
1890 19 8003 1864 (8003-2) = f₂
1864 65 8003 1824
1824 31 0008 1745 (8002) = 00000000f₂
On ne conserve de f₂ que les
chiffres de rang -18, -19 et
-20 qui seuls peuvent avoir
une influence sur L.

1745 16 1681 1940 (8003-2) = f₂ - f
1940 11 1653 1962 (8001) = 9999999999 +
1962 15 1965 1874
1874 10 8001 1990
1990 15 1814 1924 (1814) = 0000000001 +
(8003-2) = f₂ - f + 1 = L
(1731) =l₁.....
(1717) =l₂.....
(32)

Les 2 tranches du coefficient
L sont chargées en 1717 et 1730
en vue de leur utilisation par
le SP de Multiplication pour
le calcul de MxL.

5.7 Calcul du produit MxL pour la Division en Double Précision.

Les termes M et L ayant été chargés en 1715, 1730 et
1717, 1731 il est possible d'utiliser le SP de Multiplica-
tion pour calculer le produit MxL.

(32) 1854
Sous-Programme de
Multiplication
.. 1821
(12) (8003-2) =e₁.....e₂.....
La division étant terminée,
la suite est prise en 1821
à l'entrée du SP de RECADRAGE.

6° Sous-Programme de RECADRAGE.

A l'entrée dans le SP de RECADRAGE l'exposant brut du

résultat est chargé en 1700 et la mantisse brute se trouve en 8003-2.

Le but du Sous-Programme est de normaliser la mantisse sur la position 8 de 8003 afin de libérer les positions 10 et 9 qui devront contenir l'exposant du résultat.

6.1 Adaptation des résultats Simple Précision

Le SP de RECADRAGE est commun à la Simple Précision et à la Double Précision. Mais comme la mantisse d'un résultat en Simple Précision ne comporte qu'une seule tranche chargée en 8002 à l'entrée du Sous-Programme, il est nécessaire de l'adapter avant d'utiliser le SP de RECADRAGE.

(33) 1775 31 0000 1871 (8003-2)=0000000000... $a_1 \times b_1$...
 (24) 1871 60 8002 1830 (8003) = c_1
 1830 21 1715 1974 (1715) = c_1
 1974 20 1730 1941 (1730)=(8002) = 0000000000 ±
 1941 44 1845 1695 En 1695, la mantisse étant
 (36) (37) nulle, le résultat l'est aussi et il faut charger 1700 de zéros.

6.2 Normalisation de la mantisse sur la position 8 de 8003.

(12) 1821 21 1715 1975 (1715) = c_1
 1975 20 1730 1991 (1730) = c_2
 1991 45 1744 1695
 (37)
 1744 44 1845 1698
 (36) (38)
 (36) 1845 36 0000 1917 (8002) = xxxxxxxx 02
 1917 35 0008 1742 (8002) = 02 00000000
 1742 67 8002 1801
 1801 16 1914 1876 (8002) = 0(Z-2) 0000 0000
 1876 30 0004 1842 (8002) = 00 000(Z-2) 0000
 1842 45 1946 1747 En 1747, Z=2: aucun recadrage
 (39) de la mantisse n'est nécessaire
 1946 46 1800 1850

1800 69 1816 1967 (40) (8001) = 31 xxxx 1900
 Nous avons Z < 2, il faut donc décaler la mantisse vers la droite pour la normaliser

sur la position 8 de 8003.
 (8001) = 35 xxxx 1900
 Nous avons Z > 2, il faut donc décaler la mantisse vers la gauche pour la normaliser sur la position 8 de 8003.

(40) 1967 22 1653 1716 (1653) = 3^1_5 000(Z-2) 1900
 1716 30 0004 1880 (8002) = 00 0000 000(Z-2)
 1880 66 8002 1892 (8002) = 00 0000 000(Z-2)
 1892 15 1700 1866 (8002) = 00000000(Y+2-Z)
 1866 20 1700 1916 (1700) = exposant corrigé.
 1916 60 1715 1926
 1926 15 1730 1653 (8003-2)=.... c_1 c_2
 1653 3^1_5 |Z-2| 1900 (41) (8003-2)=0?.... c_1 c_2
 La mantisse normalisée est en 8003-2. La suite est prise en 1900 à l'entrée du SP de SORTIE.
 (38) 1698 67 1700 1966 (8003) = c_1 = 00000000000
 1966 16 1976 1831 (1976) = 0000000010 +
 1831 20 1700 1930 (1700)=(8002)= 000000000(Y-10)
 Puisque c_1 est nul on diminue Y de 10 unités et on reprend la suite des recadrages sur c_2 en 1821.
 1930 60 1730 1821 (12) (8003-2)=.... c_200000000000

7° Sous-Programme de SORTIE

7.1 Test sur la position 9 de 8003

A l'entrée la mantisse normalisée se trouve en 8003-2, la position 10 de 8003 étant libre, mais la position 9 pouvant ne pas l'être.
 (41) 1900 69 1719 1881 (8001) = 00 0⁸00 1650
 1881 97 1942 1992 En 1992, cas d'un résultat en Double Précision.
 (42)
 1942 31 0000 1931 (8003-2)=00000000000?.... c_1
 1931 60 8002 1992 (42) (8003-2)=0?.... c_10000000000
 Le résultat Simple Précision ayant été placé en 8003, il est possible de tester la position 9 de 8003.

A la sortie de l'Arithmétique le résultat de l'opération apparaît sous 2 formes :

Forme "2 Mots" : $(8003-2) = Y\bar{Y} \dots c_1 \dots c_2 \dots$
 et $\left\{ \begin{array}{l} (1746) = Y\bar{Y} \dots c_1 \dots \\ (1730) = \dots c_2 \dots \end{array} \right.$

Forme "3 Mots" : $\left\{ \begin{array}{l} (1700) = 00000000 Y\bar{Y} + \\ (1715) = 00 \dots c_1 \dots \\ (1730) = \dots c_2 \dots \end{array} \right.$

avec en Simple Précision $C_3 = 0000000000$

(42) 1992 21 1715 1832
 1832 20 1730 1895
 1893 36 0000 1882
 1882 67 8002 1943
 1943 16 1814 1932
 1932 45 1747 1781
 (39) (7)

(1715)=(8003)= $0\bar{Y} \dots c_1 \dots$
 (1730)=(8002)= $\dots c_2 \dots$
 (8002) = xxxxxx 0Q
 (8002) = xxxxxxx0(Q-1)
 Si la position 9 est occupée, d'une part xxx. x est nul et d'autre part Q-1 = 0, donc la suite est prise en 1781 pour reprendre le SP de RE-CADRAGE.
 Si la position 9 est libre, la suite est prise en 1747.

7.2 Tests sur l'exposant Y

L'exposant doit être un nombre positif de 2 chiffres:

00cY499

(39) 1747 65 1700 1993
 1993 45 1996 1804
 (43)
 1996 46 1695 1998
 (37)
 1998 16 1844 1851
 1851 46 1894 1944

(8002) = 00000000 Y.±
 En 1804, Y est nul. La suite est prise en 1804 pour char-1715 et 1730 de zéros.
 En 1695 $Y < 0$. On considère le résultat comme nul et on charge le PA de zéros.
 (8002) = 00000000(Y-100)
 En 1944 Y est supérieur à 99. Il y a donc overflow et on trouve l'arrêt :

01 0111 8000

7.3 Résultat sous forme "2 Mots"

1894 15 8001 1994
 1994 35 0008 1895
 1895 20 1681 1945
 1945 60 1715 1995
 1995 15 1730 1847
 1847 46 1901 1652
 1901 11 1681 1910
 1652 10 1681 1910
 (29) 1910 21 1746 1740
 1740 Inst.Sortie

(1681)=(8002)= $Y\bar{Y}00000000 +$
 (8003-2)= $00 \dots c_1 \dots c_2 \dots$
 (8003-2)= $Y\bar{Y} \dots c_1 \dots c_2 \dots$
 (8003-2)= $Y\bar{Y} \dots c_1 \dots c_2 \dots$
 (1746) = $Y\bar{Y} \dots c_1 \dots$
 En 1740 on retrouve l'instruction de Sortie de l'Arithmétique.

ANNEXE 3 ---

ARITHMETIQUE TRIPLE PRECISION A.T.P. ou B.B.L. C pour Nombres Réels , Virgule Flottante FLAIR Triple Précision

Entrées dans l'Arithmétique

1775 : Addition	TAA
1776 : Soustraction	TAS
1777 : Addition de valeur absolue	TAV
1778 : Division	TDV
1779 : Multiplication	TMU
1780 : Soustraction de valeur absolue	TSV
1781 : Transfert du P.A. vers la mémoire générale	TTR
1782 : Transfert dans le P.A.	TZA
1783 : Transfert dans le P.A. de l'opposé	TZS
1784 : Transfert dans le P.A. de la valeur absolue	TZV
1785 : Transfert dans le P.A. de l'opposé de la valeur absolue	TZW
1799 : Chargement des distances entre Mot 1 et Mot 2 - Mot 2 et Mot 3	TZZ

Macro-Code Symbolique

Sorties de l'Arithmétique

en 1740 dans tous les cas , sauf
en 8003 pour les opérations TTR et TZZ

Mémoires à contenu variable

Pseudo-Accumulateur

1700 00 0000 0000 Exposant du résultat ou du premier opé-
rande de l'opération.
1715a₁..... Mot 1 du résultat ou du premier opérán-
de.
1730a₂..... Mot 2 du résultat ou du premier opérán-
de.
1745a₃..... Mot 3 du résultat ou du premier opérán-
de.

Pseudo-Distributeur.

1705 00 0000 0000 Exposant du second opérande
1717b₁..... Mot 1 du second opérande
1731b₂..... Mot 2 du second opérande
1754b₃..... Mot 3 du second opérande.
1717, 1731 et 1754 sont également utili-
sés comme mémoires de travail par le
sous-programme de division.

Instructions à adresse facteur variable

1554 15 (P₁) 1631 Appel du Mot 2 de l'opérande dans
8002.
1574 69 (P₁) 1594 Appel du Mot 3 de l'opérande dans 8003
1596 60 (P₁) 1561 Appel du Mot 1 de l'opérande dans 8003
1606 31 oodu 1890 oodu = |a-b|, différence des exposants
et nombre des décalages à droite avec
arrondi à effectuer sur l'une des 2 man-
tisses.
autre instruction de même type.
1639 31 oodu 1950 oodu = |a-b|, différence des exposants
et nombre de décalages à droite à ef-
fectuer sur l'une des 2 mantisses.
1759 30 oodu 1850 Recadrage vers la gauche de la mantis-
se du résultat. Le nombre des décalages,
(Z-3) est calculé par le Sous-Programme.
1826 35 [Z-3] 1674 Transfert du Mot 2 du résultat de l'opé-
ration vers la mémoire P₁.
1846 24 (P₁) 1993 Instruction de même type que celle en
1759.
1859 30 oodu 1602 Instruction de même type que celle con-
tenue en 1826, le décalage se faisant
vers la droite.
1872 30 [Z-3] 1645 Instruction de même type que celle con-
tenue en 1826
1876 35 [Z-3] 1645

1893 24 (P₁) 8003 Transfert du Mot 3 du résultat de l'opé-
ration vers la mémoire P₁.
1975 31 [Z-3] 1968 Instruction de même type que celle con-
tenue en 1872; le décalage se faisant
avec arrondi.
1992 24 (P₁) 1943 Transfert du Mot 3 du résultat de l'opé-
ration vers la mémoire P₁.
1732 00 D₁ 0000+ Distance entre les Mots 1 et 2 des nom-
bres traités.
1747 00 D₁ 0000+ Distance entre les Mots 2 et 3 des nom-
bres traités.

Mémoires de Travail.

Ce sont les mémoires :

1561, 1587, 1594, 1603, 1631, 1710, 1746 et 1756.

Ces mémoires sont utilisées lors des transferts provisoires
à l'intérieur des différents Sous-Programmes, de sorte que
leurs contenus sont essentiellement variables au cours du
déroulement des instructions.

Cependant 3 mémoires jouent également un rôle plus parti-
culier.

1587 { 00 0800 1652 Orientation 9 : Transfert vers le P₁
00 0900 1652 { 9 : Opération arithmétique.
1594 Orientation On y charge la première instruction du
Sous-Programme Opérationnel sélectionné
à l'entrée.
1746 yyy....c₁... Mot 1 du résultat à la sortie de l'A-
rithmétique.

1740 (Inst.Sortie) Instruction à prendre à la suite de
l'opération en cours.

Mémoires à contenu fixe.

1581 00 0800 1652 Orientation vers le sous-programme de
multiplication ou vers les opérations
de transfert vers le P₁.
1582 00 0900 1652 Orientation vers le sous-programme de
division ou vers les opérations arith-
métiques.
1691 01 0333 8000 L'une des distances D₁ ou D₂ est nulle
ou bien n'a pas été chargée en 1732
ou 1747.
1696 01 0111 8000 Arrêt Overflow : l'exposant du résultat
est devenu supérieur à 999.

1810 01 0222 8000 Le programme aurait à effectuer une division dont le diviseur est nul.

1625 00 0000 0001 -
1626 00 0000 0001 +
1643 00 0010 0000 +
1680 00 0000 1000 +
1695 00 0000 0002 +
1722 03 0000 0000 +
1815 00 0007 0000 +
1844 00 0100 0000 +
1863 00 0000 0498 +
1878 00 0000 0010 +
1881 00 0000 0020 +
1908 00 0000 0503 +
1971 99 9999 9999 +

1° Sous-Programme d'ENTREE

L'entrée dans l'Arithmétique est spécifique de chaque opération ; on distingue 6 opérations arithmétiques, 4 opérations de transfert vers le PA et une opération de transfert vers la mémoire générale. A l'entrée, les contenus de 8003 et 8002 sont les suivants :

(8003) = Instruction de Sortie

(8002) = 00 F xxxx

F étant l'adresse du Mot 1 du nom-bre traité par l'opération.

1775 69 1635 1591
1776 69 1636 1591
1777 69 1637 1591
1778 69 1581 1591
1779 69 1582 1591
1780 69 1638 1591
1782 69 1585 1641
1783 69 1636 1641
1784 69 1637 1641
1785 69 1638 1641

On charge 8001 de la première instruction du SPO correspondant à l'opération.

On charge 8001 de la première instruction du sous-programme de transfert correspondant à l'entrée.

1591 24 1594 1597
1597 69 1581 1594

(8001) transféré en 1594

(8001) = 0009001652+ : un 9 en position 7 : opération arithmétique.

1641 24 1594 1647
1647 69 1582 1584
1584 24 1587 1590
1590 21 1740 1593
1593 69 1596 1599
1599 22 1746 1649
1649 65 1732 1687
1687 45 1640 1691
1640 10 1747 1551
1551 44 1555 1691
1555 15 1746 1601
1601 69 1554 1557
1557 22 1561 1564
1564 15 8003 1571
1571 69 1574 1577
1577 22 1631 1746
1746 60 F 1561
1561 15 F 1631
1631 69 F 1594
(8001) transféré en 1594
(8001) = 0008001652+ : 8 en position 7 : opération de transfert vers le PA.
(1587) = 00 0900 1652
(1740) = Instruction de Sortie.
(8002) = 00 F xxxx
(8001) = 60 F 1561
Instruction d'appel du Mot 1 transférée en 1746.
(8002) = 00 D' 0000
(1691) = 01 0333 8000
La distance D' étant nulle ou bien n'ayant pas été chargée, il y a arrêt du déroulement du programme.
(8003) = 00 D" 0000
1691 : voir ci-dessus.
(8002) = 60 (F+D') 1561

(8001) = 15 (F+D') 1631
Instruction d'appel du Mot 2 transférée en 1561.

(8002) = 60 (F+D'+D") 1561

(8001) = 69 F" 1594
Instruction d'appel du Mot 3 transférée en 1631.

(8003) = p p p...b₁... = Mot 1

(8002) =b₂.... = Mot 2

(8001) =b₃.... = Mot 3

En 1594 on retrouve la première instruction du SPO ou du SPT sélectionné à l'entrée.

Chargement des distances D' et D"

Les distances D' et D" doivent être chargées en position d'adresse facteur dans les mémoires 1732 et 1747. A l'entrée dans l'Arithmétique, les contenus de 8003 et 8002 sont les suivants :

(8003) = Instruction de Sortie

(8002) = 00 D' D" +

1799 21 1740 1694
1694 60 8002 1744
1744 30 0004 1794
(1740) = Instruction de Sortie
(8003) = 00 (DDDD)' (DDDD)' = 199
(8003) = '00 0000 (DDDD)'

Transfert de - B : (1594) = (1636)

(2) 1636 24 1746 1849 (1746)=(8001)=...b₃...
 1849 20 1603 1556 (1603)=(8002)=...b₂...
 1556 21 1561 1664 (1561)=(8003)=...b₁...
 1664 61 1746 1701
 1701 21 1754 1607 (1754)=-(...b₃...)
 1607 61 1561 1615 (8003)=-(...b₁...)
 1615 16 1603 1657 (8002)=-(...b₂...)
 1657 69 1587 1690 (8001)=00 0800 1652
 1690 97 1550 1595
 (3)
 1550 69 1754 1585 (1) (8001)=-(...b₃...)
 La suite des instructions est prise en 1585 après changement du signe de B.

Transfert de + B : (1594) = (1637)

1637 46 1636 1635 Si B est positif, la suite est prise en 1635: Addition.
 (2) (1') Si B est négatif, la suite est prise en 1636: Soustraction.

Transfert de - B : (1594) = (1638)

1638 46 1635 1636 Si B est positif, la suite est prise en 1636: Soustraction.
 (1') (2)
 (1') 1635 24 1754 1657 Si B est négatif, la suite est prise en 1635: Addition.
 1657 ... (1754)=(8001)=...b₃...
 ... 1585 (1) Voir ci-dessus.

2.2 Transfert vers la mémoire Générale

Le résultat de l'opération étant chargé sous la forme 3 Mots dans les mémoires 1730, 1745 et 1746 :
 (1746) = YV...C₁... (1730) = ...C₂...
 et (1745) = ...C₃...
 il doit être transféré dans les mémoires d'adresses F, F' (F+D') et F'' (F+D'+D''), F étant l'adresse indiquée en position d'adresse facteur de 8002.
 A l'entrée : (8003) = Instruction de Sortie
 (8002) = 00 F XXX

(8002) = (DDDD)' = 00 0000
 (8003) = (DDDD)' = 00 0000
 (8002) = (DDDD)' = 00 0000
 (8001) = 00 0100 0000+
 (8003) = 00 (DDDD)' = 0000 transféré en 1747
 (8002) = 00 (DDDD)' = 0000 transféré en 1732
 En 1740, on retrouve l'Instruction de Sortie.

2° Sous-Programmes de TRANSFERT

2.1 Transferts vers le PA

A l'entrée dans les Sous-Programmes de Transfert vers B PA, 8001, 8002 et 8003 contiennent le nombre B sous la forme 3 Mots :

(8001) = ...b₃...
 (8003) = ...b₁... (8002) = ...b₂...
 et la mémoire d'orientation 1587 contient :
 (1587) = 00 0800 1652

Transfert de B: (1594)=(1585)

(1) 1585 24 1745 1598 (1745) = ...b₃...
 1598 21 1746 1699 (1746) = ...b₁...
 1699 20 1730 1583 (1730) = ...b₂...
 1583 30 0007 1749
 1749 21 1561 1614 (1561) = (8002) = 00000000...
 1614 65 8002 1573 (8002) = ...b₁...
 1573 30 0003 1681
 1681 20 1715 1568 (1715) = (8002) = 000...b₁...
 1568 67 1561 1565
 1565 20 1700 1553 (1700) = (8002) = 00000000...
 1553 65 1730 1685 (8002) = ...b₂...
 1685 10 1746 1651 (8003) = ...b₁...
 1651 69 1745 1740 (8001) = ...b₃...
 1740 Inst. de Sortie.

1781 69 1992 1895 (8002) = 00 F xxxx
 1895 22 1756 1682 (1756)=(8001)= 24 F 1943
 1882 15 1732 1793 (8002) = 00 (F+D') xxxx
 1793 69 1846 1832
 1832 22 1631 1843 (1631)=(8001)= 24 F' 1993
 1843 15 1747 1882 (8002) = 00 (X+D'+D'') xxxx
 1882 69 1893 1896
 1896 22 1603 1932 (1603)=(8001) = 24 F'' 8003

1932 69 1746 1756
 1756 24 F 1943 (8001) = YY'...0...
 Transfert du Mot 1 en F
 1943 69 1730 1631 (8001) =b₂....
 1631 24 F' 1993 Transfert du 2 Mot 2 en F'
 1993 69 1745 1603 (8001) =c₃....
 1603 24 F'' 8003 Transfert du 3 Mot 3 en F''
 8003 Instruction
 de Sortie.

3° Sous-Programme d'ADDITION

A l'entrée dans le Sous-Programme, le second opérande B est chargé dans 8003-2-1 sous la forme :

(8001) =b₃....
 (8003-2) = βββ...b₁...b₂....
 et puisqu'il s'agit d'une opération arithmétique, 1587 contient un 9 en position 7 :
 (1587) = 00 0900 1652

3.1 Opérations se ramenant à l'Addition

3.1.1 SOUSTRACTION: (1594)=(1636)

(2) 1636 24 1746 1849

 1657 69 1587 1690 (8001) = 00 0900 1652

1690 97 1550 1595
 (3)

(3) 1595 20 1731 1634 (1731)=(8002)=b₂....
 Le signe de B ayant été changé, la suite des instructions est prise dans le sous-programme d'addition.

3.1.2 Addition de VALEUR ABSOLUE: (1594)=(1637)

1637 46 1636 1635 Si B est positif, la suite est prise en 1635: Addition
 (2) (1') Si B est négatif, la suite est prise en 1636: Soustraction.

3.1.3 Soustraction de VALEUR ABSOLUE: (1594)=(1638)

1638 46 1635 1636 Si B est positif, la suite est prise en 1636: Soustraction.
 (1') (2) Si B est négatif, la suite est prise en 1635: Addition.

(1') 1635 24 1754 1657 (1754)=(8001)=b₃....
 1657 69 1587 1690 (8001) = 00 0900 1652
 1690 97 1550 1595
 (3)

3.2 ADDITION: (1594)=(1635)

A l'entrée dans le Sous-Programme d'Addition, le premier opérande se trouve déjà chargé dans le PA sous la forme 4
 Mots : { (1700) = 0000000000 , (1715) = 000...a₁...
 (1730) =a₂.... et (1745) =a₃....
 Le second opérande se trouve dans 8003-2-1 sous la forme 3 Mots, après un éventuel changement de signe.

3.2.1 Chargement de B dans le PD.

1635 24 1754 1657 (1754)=(8001)=b₃....
 1657 69 1587 1690 (8001) = 00 0900 1652
 1690 97 1550 1595
 (3) (1731)=(8002)=b₂....
 (3) 1595 20 1731 1634

1634 30 0007 1751
1751 21 1746 1899
1899 65 8002 1707
1707 30 0003 1665
1665 20 1717 1570
1570 67 1746 1801
1801 20 1705 1558

$(1746)=(8003)=0000000|\beta\beta\beta|$
 $(8002) = \dots b_1 \dots 000$
 $(1717)=(8002)=000\dots b_1\dots$

$(1705)=(8002)=0000000|\beta\beta\beta| +$
Le second opérande B est donc chargé dans le PD sous la forme 4 Mots :

$(1705) = 0000000|\beta\beta\beta| +$
 $(1717) = 000\dots b_1\dots$
 $(1731) = \dots b_2 \dots$
 $(1754) = \dots b_3 \dots$

3.22 Etude des cas particuliers.

1558 45 1562 1563
(4)

$(8002) = 0000000|\beta\beta\beta| +$
En 1563, le nombre B est nul; il ne reste plus qu'à charger a_1 et a_2 dans 8003-2 avant de passer au Sous-Programme de RECADRAGE.
 $(8002) = 0000000|\alpha\alpha\alpha| +$

1562 67 1700 1605
1605 45 1608 1559
(5)

$(8003) = 00000000000 +$

1559 69 1705 1658
1658 24 1700 1653
(12) 1653 69 1754 1757

$(8003-2) = 00000000000 \dots 00 +$
 $(1700)=(8001)=0000000|\beta\beta\beta| +$

1757 24 1745 1648
1648 60 1717 1621
1621 15 1731 1735
(6)

$(1745)=(8001)=\dots b_3 \dots$
 $(8003) = 000\dots b_1 \dots$
 $(8002) = \dots b_2 \dots$

L'opérande A étant nul, B est partiellement chargé dans le PA et en 8003-2. La suite est prise en 1735 à l'entrée du SP de RECADRAGE.

3.23 Détermination de l'exposant

(5) 1608 18 1705 1609
1609 45 1612 1613
(7)

$(8002) = 0000000|\alpha.\beta|$

1612 35 0004 1723
1723 46 1676 1627
(8) (9)

$(8002) = 00\ 00(\alpha-\beta)\ 0000$
En 1613 $\alpha-\beta=0$, donc $\gamma=\alpha=\beta$
En 1676 $\alpha-\beta<0$, donc $\gamma=\beta$
En 1627 $\alpha-\beta>0$, donc $\gamma=\alpha$

3.24 Positionnement relatif des mantisses

1er Cas: $\gamma=\alpha=\beta$

Il n'y a aucun décalage à effectuer sur l'une ou l'autre des mantisses puisque

$$(\gamma-\alpha) = (\gamma-\beta) = 0$$

On passe directement au § 3.25 de l'addition des mantisses positionnées.

2ème Cas: $\gamma=\beta$

On commence par fractionner le nombre de décalages en tranches de 10 au maximum puis on recharge la mantisse à décaler dans le PA avant de passer à l'addition.

(8) 1676 69 1705 1708

$$(1700)=(8001)=0000000|\beta\beta\beta|$$

$$(8002) = 00\ 0000\ 0000$$

$$(1603)=(8001)=31\ 0000\ 1800$$

$$(1587)=(8001)=30\ 0000\ 1850$$

$$(1643) = 00\ 0010\ 0000 +$$

$$(8002) = 00\ 0(\alpha-\beta+10)\ 0000$$

1697 46 1750 1901

(10)

1750 15 8001 1809

1809 46 1712 1663

(11)

1712 15 1815 1769

1769 46 1653 1773

(12) (13)

1901 60 1750 1686

1686 15 1745 1603

1603 31 0000 1800

1800 21 1730 1733

1733 20 1745 1798

1798 60 1715 1587

1587 30 0000 1850

1850 15 1750 1736

1736 21 1715 1768

1768 20 1730 1613

(7)

Nous avons $\beta-10 \leq \alpha \leq \beta$
Donc une seule tranche de 0000 décalages sur la mantisse α .
 $(8003-2) = \dots a_2 \dots \dots \dots a_3 \dots$
 $(8003-2) = 000 \dots a_2 \dots \dots a_3 \dots$
 $(1730)=(8003) = \dots a_3 \dots$
 $(1745)=(8002) = 000 \dots a_2$
 $(8003) = 000\dots a_1\dots$

$(8003-2)=00000 \dots a_1 \dots a_2 \dots$
 $(1715)=(8003)=00000\dots a_1$
 $(1730)=(8002)=\dots a_2 \dots$

(11) 1663 60 1715 1819

Nous avons: $\beta - 20 \leq \alpha < \beta - 10$
On néglige a_3 , a_1 et a_2 ayant
pris la place de a_2 et a_1 en
effectue le décalage de $3u$
positions.

1819 20 1715 1718

(1715)=(8002)=0000000000

1718 15 1730 1603

(1715)=(8002)=000... a_1 a_2

(12) 1653 69 1754 1757

Nous avons: $\alpha < \beta - 27$
 λ est donc considéré comme
nul devant B. la suite est
prise en 1653 pour le trans-
fert de B dans le PA.

.... .. 1735

(6)

(13) 1773 65 1715 1869

Nous avons: $\beta - 27 \leq \alpha < \beta - 20$
On néglige a_3 et a_2 , a_1 ayant
pris la place de a_3 on effec-
tue le décalage de $3u$ po-
sitions.

1869 21 1715 1603

(8003-2)=0000000000 000... a_1 ...
(1715) = 0000000000

3ème Cas : $\gamma = \alpha$

On commence par fractionner le nombre des décalages en tran-
ches de 10 au maximum puis on recharge la mantisse b décalée
dans le PA avant de passer à l'addition.

(9) 1627 66 8002 1786

(8002) = -(00 oodu 0000)
(oodu = $|\alpha - \beta|$)

1786 69 1639 1592

(1603)=(8001)= 31 oodu 1950

1592 22 1603 1706

(1587)=(8001)= 30 oodu 1602

1706 69 1859 1762

(8002) = 00 0($\beta - \alpha + 10$) 0000

1762 22 1587 1840

(8002) = 00 0($\beta - \alpha + 20$) 0000

1840 15 1643 1797

(8002) = 00 0($\beta - \alpha + 27$) 0000

1797 46 1900 1552

(8002) = 00 0($\beta - \alpha + 27$) 0000

1900 15 8001 1909

1909 46 1812 1713

1713 15 1815 1919

1919 46 1563 1823

(4) (17)

(15) 1552 60 1731 1836

Nous avons: $\alpha - 10 \leq \beta < \alpha$
On effectue une tranche de u
décalages sur la mantisse b.

1836 15 1754 1603

(8003-2)=... b_2 b_3

(18) 1603 31 oodu 1950

(8003-2)=00... b_2 b_3 ..

1950 21 1731 1684

(1731)=(8003)= 00... b_2

1684 20 1754 1807

(1754)=(8002)= ... b_3

1807 60 1717 1587

(8003) = 000... b_1 ..

1587 30 oodu 1602

(8003) = 000.0 .. b_1

1602 15 1731 1886

(8003-2)=000.0 .. b_1 b_2

1886 21 1717 1620

(1717)=(8003)= 000.0 .. b_1

1620 20 1731 1613

(1731)=(8002)= ... b_2

(7)

(16) 1713 60 1717 1721

Nous avons: $\alpha - 20 \leq \beta < \alpha - 10$
On néglige b_3 , b_1 et b_2 ayant
pris la place de b_2 et b_3 on
effectue le décalage de $3u$
positions.

1721 20 1717 1670

(1717)=(8002)=0000000000

1670 15 1731 1603

(8003-2)=000... b_1 b_2

(18)

(4) 1563 60 1715 1719

Nous avons: $\beta < \alpha - 27$
B est donc considéré comme
nul devant A. On transfère
 a_1 et a_2 dans 8003-2

1719 15 1730 1735

En 1735 on entre dans le Sous-
Programme de RECADRAGE.

(17) 1823 65 1717 1771

Nous avons: $\alpha - 27 \leq \beta < \alpha - 20$
On néglige b_3 et b_2 , b_1 ayant
pris la place de b_3 on effec-
tue le décalage de $3u$ posi-
tions.

1771 21 1717 1603

(8003-2)=0000000000 000... b_1 ..

(18)

Les mantisses ayant été positionnées l'une par rapport
à l'autre et la mantisse décalée ayant été rechargée soit
dans le PA, soit dans le PD, il ne reste plus qu'à effectuer
la sommation des contenus du PA et du PD.

3.25 Sommation des contenus du PA et du PD

(7) 1613 65 1745 1949

(8002) = $a_3 + b_3 = r_3$

1949 15 1754 1659

(1745) = (8002) = r_3

1659 20 1745 1698

1698 65 8003 1655 (8002) = 000000000⁰
report de la sommation précédente.

1655 15 1730 1835 (8003-2) = report + $a_{1,2} + b_{1,2}$
= $x_{1,2}$

1835 15 1731 1885 (1730) = (8002) = x_2

1885 10 1715 1569 (1715) = (8003) = x_1

1569 10 1717 1671

1671 20 1730 1633

1633 21 1715 1618

Vérification des signes des résultats partiels

1618 45 1572 1623 En 1623, le résultat se réduit à x_2 . La vérification est sans objet. La suite est prise à l'intérieur du SP de RECADRAGE.

1572 44 1575 1576 En 1576, $r_1 = 0$ mais x_2^{40}

1575 60 1715 1619 (8003) = x_1

1576 60 1730 1619 (8003) = x_2

1619 19 1745 1765 (8003-2) = $x_1 x_3$ ou $x_2 x_3$

1765 46 1668 1563 En 1563, le produit est positif cela signifie que les 2 résultats partiels sont de même signe: aucune correction n'est nécessaire.

1668 65 1730 1935 En 1668, le produit étant négatif, cela signifie que x_2 n'a pas le même signe que $x_{1,2}$. Il faut donc corriger le premier.

1935 45 1588 1589 (8002) = x_2

1589 10 1715 1669 (8003-2) = $\dots x_1 \dots$ 0000000000

1669 46 1622 1673 On recherche le signe du résultat.

1622 60 1625 1579 (8003) = (1625) = 0000000001 -

1673 60 1626 1579 (8003) = (1626) = 0000000001 +

1579 21 1746 1999 (1746) = 0000000001 ±

1999 15 1745 1600 (8002) = x_3 corrigé = e_3

1600 20 1745 1748 (1745) = $\dots e_3 \dots$

1748 66 1746 1851 (8002) = 0000000001 +

1851 10 1715 1719

1719 15 1730 1735 (6) (8003-2) = $x_{1,2}$ corrigé = 000... $e_1 \dots \dots e_2 \dots$

Le résultat étant corrigé, la suite est prise en 1735 à l'entrée du SP de RECADRAGE.

(20) 1588 60 1730 1586 (8003) = x_2
 x_2 n'étant pas nul, on s'en sert pour corriger x_3 .

1586 15 1745 1650 (1730) = (8003) = x_2 corrigé =

1650 21 1730 1683 $e_2 \dots$

1683 20 1745 1563 (4) (1745) = (8002) = x_3 corrigé = $e_3 \dots$

A la sortie du Sous-Programme d'ADDITION, l'exposant γ est chargé en 1700. La tranche e_3 est en 1745 et les tranches e_1 et e_2 sont en 8003-2. Ces tranches ont même signe. La suite des instructions est prise en 1735, à l'entrée du SP de RECADRAGE pour une éventuelle normalisation de la mantisse.

4° Sous-Programme de MULTIPLICATION

4.1 Transfert de B dans le PD
Recadrage de la mantisse a

Ces instructions sont communes aux Sous-Programmes de Multiplication et de Division. Dans les 2 cas, la mantisse b doit être chargée sous la forme suivante :

(1717) = $b_1 \dots$ (1731) = $b_2 \dots$ (1745) = $b_3 \dots$ 000

Dans le cas de la multiplication:

(1715) = $a_1 \dots$ (1730) = $a_2 \dots$ (1745) = $a_3 \dots$ 000

Dans le cas de la division:

(1715) = 0..... $a_1 \dots$ (1730) = $a_2 \dots$ (1745) = $a_3 \dots$ 00

A l'entrée, le second opérande B est chargé en 8003-2-1 sous la forme:

(8001) = $b_3 \dots$
(8003-2) = $b_1 \dots$ $b_2 \dots$

et l'on a:

(1594) = 00 0800 1652 : multiplication
(1594) = 00 0900 1652 : division

1594 00 0⁹00 1652
 1652 24 1754 1857
 1857 20 1731 1734
 1734 30 0004 1702
 1702 21 1756 1560
 1560 20 1717 1720
 1720 60 1731 1936
 1936 15 1754 1610
 1610 35 0003 1969
 1969 21 1731 1834
 1834 20 1754 1907
 1907 60 1715 1770
 1770 15 1730 1737
 1737 69 1594 1847
 1847 97 1752 1802
 1752 35 0003 1611
 1802 35 0002 1611
 1611 21 1715 1818
 1818 20 1730 1833
 1833 65 1745 1852
 1852 69 1594 1897
 1897 97 1902 1753
 1902 35 0003 1661
 1753 35 0002 1661
 1661 10 1730 1787
 1787 20 1745 1848
 1848 21 1730 1883

4.2 Etude des cas particuliers.

1883 67 1700 1755
 1755 45 1758 1660
 (21)
 1758 67 1756 1711
 1711 20 1705 1808
 1808 69 1594 1947
 1947 97 1803 1853
 (22)

(1754)=(8001)=... b_3 ...
 (1731)=(8002)=... b_2 ...
 (1756)=(8003)=00000000 $\beta\beta\beta$
 (1717)=(8002)=... b_1 ...
 (8003-2)=... b_2 ... b_3 ...
 (1731)=(8003)=... b_2 ...
 (1754)=(8002)=... b_3 ...000
 (8003-2)=000... a_1 ... a_2 ...
 (8001)=00 0⁸00 1652
 (8003-2)=... a_1 ... a_2 ...000
 (8003-2)=0... a_1 ... a_2 ...00
 (1715)=... a_1 ...
 (1730)=... a_2 ... a_3 ...000
 (8002)=... a_3 ...
 (8001)=00 0⁹00 1652
 (8003-2)=00000000... a_3 ...000
 (8003-2)=00000000... a_3 ...00
 (8003-2)=... a_2 ... a_3 ...000
 (1745)=... a_3 ... a_3 ...000
 (1730)=... a_2 ...

(8002) = 00000000 α
 En 1660, le multiplicande ou le dividende étant nul, le produit ou le quotient l'est également et donc déjà chargé dans le PA. La suite est prise à la fin du SP de SORTIE.
 (8002) = 00000000 β
 (1705) = 00000000 β
 (8001) = 00 0⁸00 1652
 Séparation des SP de Multiplication : 8 en position 7 et de division : 9 en position 7.

1803 45 1806 1858
 (22) (29)
 (29) 1858 21 1700 1742
 (30) 1742 21 1715 1931
 1931 21 1730 1942
 1942 21 1745 1660 (21)

4.3 Calcul de l'exposant Y.

(22) 1806 17 1700 1805
 1805 16 1908 1763
 1763 20 1700 1903

(8002) = 00000000(β + α) +
 (8002) = 00000000(β + α -503)
 (1700) = 00000000 γ
 On charge dans 1700 l'exposant du résultat diminué de 3 unités en raison de la correction d'exposant qui aura lieu dans le SP de RECADRAGE.

4.4 Calcul de la mantisse c.

4.4.1 Calcul et transfert des produits partiels

Dans ce calcul on négligera les produits $a_3 \times b_2$, $a_2 \times b_3$ et $a_1 \times b_1$ introduisant par là même une erreur qui reste en valeur absolue inférieure à 3.10^{-10} alors que l'erreur introduite lors du recadrage de la mantisse résultant est de l'ordre de $0.5.10^{-2}$ soit 500.10^{-10} , ce qui rend la première négligeable.

(23) 1903 60 1715 1820
 1820 19 1717 1837
 1837 21 1603 1856
 1856 20 1587 1890
 1890 60 1730 1887
 1887 19 1717 1937
 1937 21 1561 1714
 1714 20 1746 1604
 1604 60 1745 1654
 1654 19 1717 1987
 1987 21 1594 1997
 (8003-2) = $a_1 \times b_1$
 (1603) = $c(a_1 \times b_1)$
 (1587) = $D(a_1 \times b_1)$
 (8003-2) = $a_2 \times b_1$
 (1561) = $c(a_2 \times b_1)$
 (1746) = $D(a_2 \times b_1)$
 (8003-2) = $a_3 \times b_1$
 (1594) = $c(a_3 \times b_1)$. On néglige la tranche de droite. Erreur inférieure à 1.10^{-10} .
 (8003-2) = $a_1 \times b_2$
 (1710) = $c(a_1 \times b_2)$

En 1858 le multiplicateur étant nul, le produit l'est également. Le PA est chargé de zéros.

(8003) = 00000000000 +

1813 20 1631 1884
1884 60 1730 1688
1688 19 1731 1804
1804 21 1756 1760

1760 60 1715 1920
1920 19 1754 1675
1675 31 0000 1854

$$(1631) = B(a_1 \times b_2)$$

$$(8003-2) = a_2 \times b_2$$

$$(1756) = G(a_2 \times b_2)$$

On néglige la tranche de dixième : erreur inférieure à 1.10

$$(8003-2) = a_1 \times b_3$$

$$(8002) = G(a_1 \times b_3)$$

De la tranche de droite on ne conserve qu'un report éventuel sur celle de gauche: erreur inférieure à 0,5.10⁻⁴

4.42 Calcul de la tranche c_3

Le calcul de la tranche c_4 est inutile du fait de l'erreur introduite lors du recadrage de la mantisse.

1854 15 1756 1761
1761 15 1631 1738
1738 15 1594 1904
1904 15 1746 1855
1855 20 1745 1898

$$(8003) = 000000000 \text{ report}$$

$$(1745) = (8002) = \dots c_3 \dots$$

4.43 Calcul des tranches c_2 et c_1

$$(8002) = \text{report}$$

1898 65 8003 1905
1905 15 1587 1741
1741 15 1561 1865
1865 15 1710 1915
1915 10 1603 1735 (6)

Les 2 premières tranches de la mantisse sont chargées en 8003-2, l'exposant pré-corrigé en 1700 et c_3 en 1745. La suite est prise en 1735 à l'entrée du SP de RECADRAGE. L'erreur totale introduite en négligeant certains termes de l'expansion de multiplication reste inférieure à 5,5.10⁻¹⁰

5° Sous-Programme de DIVISION

Le contenu de 1594 est alors

$$(1594) = 00\ 0900\ 1652$$

Après le chargement de B dans le PB, le recadrage de la mantisse a avec un zéro en tête et l'étude du cas particulier où $d' = 0$, puisque 1594 contient un 9 en position 7, la suite des instructions est prise en 1853.

5.1 Cas particulier $d' = 0$

$$(22) \ 1853 \ 45 \ 1906 \ 1810$$

(8002) = 0000000 β +
En 1810, un arrêt programmé interrompt le déroulement du programme puisque le diviseur se trouve nul.

$$(1810) = 01\ 0222\ 8000$$

5.2 Calcul de l'exposant γ

1906 67 1700 1860
1860 18 1705 1910
1910 15 1863 1567
1567 20 1700 1811

$$(8002) = 0000000 \alpha +$$

$$(8002) = 0000000 (\alpha - \beta)$$

$$(8002) = 0000000 (\alpha - \beta - 498)$$

(1700) = 0000000 γ
On charge en 1700 l'exposant diminué de 2 unités en raison de la correction d'exposant qui doit avoir lieu dans le SP de RECADRAGE et du décalage systématique de 2 de une position vers la droite.

5.3 Calcul de M

1811 60 1715 1970
1970 15 1730 1788
1788 14 1717 1677
1677 20 1715 1868
1868 60 8003 1725
1725 15 1745 1861
1861 14 1717 1727
1727 20 1730 1933
1933 60 8003 1791
1791 64 1717 1827
1827 20 1745 1948

$$(8003-2) = 0 \dots a_1 \dots \dots a_2 \dots$$

$$(1715) = (8002) = \dots m_1 \dots$$

$$(8003) = \text{reste de la division}$$

$$(8002) = \dots a_3 \dots 00$$

$$(1730) = (8002) = \dots m_2 \dots$$

$$(8003) = \text{reste de la division}$$

$$(1745) = (8002) = \dots m_3 \dots$$

Les 3 tranches de M sont chargées en 1715, 1730 et 1745 en vue de l'utilisation du SP de Multiplication pour le calcul du produit MxL.

5.4 Calcul du terme correctif L

5.4.1 Existence et calcul de f

1948 60 1731 1838
1838 15 1754 1911
1911 45 1764 1863 (4)

1764 65 8003 1821
1821 14 1717 1877
1877 20 1561 1814
1814 60 8003 1871
1871 15 1754 1961
1961 14 1717 1927
1927 20 1594 1998
1998 60 8003 1862
1862 64 1717 1578
1578 20 1710 1913

(8003-2)=...b₂.....b₃..000
En 1563, b₂ et b₃ sont nuls. Il est donc inutile de calculer f. La division est terminée et la suite est prise au SP de RECADRAGE
(8002) = ...b₂....
(1561)=(8002)= 000000000f'
(8003) = reste de la division
(8002) = ...b₃..000
(1594)=(8002)=...f'....
(8003) = reste de la division
(1710)=(8002)= ...f'....

5.4.2 Calcul de f² et -f³

1913 10 1594 1912
1912 31 0001 1921
1921 21 1756 1962
1962 20 1631 1934
1934 60 1561 1965
1965 35 0009 1888
1888 10 1756 1963
1963 21 1756 1864

(8003)=...f'....
(1756)=(8003)=0....f'....
(1631)=(8002)=...f'....
(8003)=000000000f'
(8003)=f'0000000000
(8003)=f'...f'.....f₁....
(1756)=...f₁....
Ayant réduit f à 2 tranches de 10 chiffres, il est possible de calculer f² et f³ dont on ne retiendra que les chiffres ayant une influence sur L.
(8003-2)= f²

1938 21 1603 1914 (1603) = G(f₁²)
1914 20 1587 1940 (1587) = D(f₁²)
1940 60 1631 1988 (8003) = ...f₁'....
1988 19 1756 1628 (8003-2) = f₁x f₁'
1628 21 1746 1964 (1746) = (8003) = G(f₁x f₁')
1964 61 1603 1566 (8003) = -G(f₁²)
1566 19 1756 1678 (8003-2) = -G(f₁²)x f₁' = -f³

5.4.3 Calcul de la somme L = 1-f+f²-f³

1678 31 0009 1616 (8003-2)=000000000.... -f³....
1616 15 1746 1666 (1746) = G(f₁x f₁')
1666 15 8001 1825
1825 15 1587 1841 (1587) = D(f₁²)
1841 10 1603 1716 (8003) = G(f₁²)
(8003-2) = f² - f³
(8003-2) = partie utile de f²-f³
1716 31 0008 1689 (1710) = ...f₁'....
1689 16 1710 1766 (8002) = tranche n° 3 de L
1766 10 1626 1831 (1754) =l₃....
1831 20 1754 1816 (8002) = report
1816 65 8003 1873 (1594) =f₁'....
1873 16 1594 1866 (1561) = 000000000f'
1866 11 1561 1916 (1971) = 9999999999 +
1916 15 1971 1875 (8003-2)=...l₁.....l₂....
1875 10 8001 1739 (1717) =l₁....
1739 21 1717 1672 (1731) =l₂....
1671 20 1731 1903 (23)

Les 3 tranches du coefficient L sont chargées en 1717, 1731 et 1754 en vue de leur utilisation par le SP de Multiplication pour le calcul du produit MxL.

5.5 Calcul du produit MxL

(23) 1903
Sous-Programme de MULTIPLICATION
..... 1735 (6) (8003-2)=...c₁.....c₂....
La tranche c₂ étant chargée en 1745, la suite est prise en 1735 à l'entrée du SP de RECADRAGE.

A l'entrée dans le SP de RECADRAGE, l'exposant brut est chargé en 1700 et la mantisse brute en 1745 (tranche n° 3) et 8003-2 pour les tranches n° 1 et 2.

Le but du Sous-Programme est de normaliser la mantisse en la recadrant sur la position 7 de 8003, les positions 10, 9 et 8 étant libérées afin d'y charger l'exposant du résultat.

(6) 1735 21 1715 1918
1918 20 1730 1789
1789 44 1693 1644

(24)

(8002) = xxxxxxxx02

(8002) = 02 0000 0000 +
(8002) = 0(Z-3) 0000 0000

En 1889, $X = 3$, donc la mantisse est déjà normalisée et aucun recadrage n'est nécessaire. La suite est prise en 1889 à l'intérieur du SP de SORTIE

(25)

1632 30 0004 1743
1743 20 1756 1667
1667 66 8002 1925
1925 30 0004 1939
1939 15 1700 1767
1767 20 1700 1817

(1700)=(8002)= 00000000(3-Z+Y)
L'exposant corrigé du fait du recadrage est chargé en 1700.

(8002) = 00 000(Z-3) 0000

1817 65 1756 1867
1867 46 1772 1822

(26) (27)

(26) 1772 69 1975 1828

Nous avons $Z=3$, il faut donc décaler la mantisse vers la droite pour libérer les positions 10, 9 et 8 de 8003.

(8001) = 31 xxxx 1968

(1594) = 31 000 |Z-3| 1968

(1710) = 30 000 |Z-3| 1645

(8003-2)=.....c₂.....c₃....

1828 22 1594 1917
1917 69 1872 1726
1726 22 1710 1967
1967 60 1730 1989
1989 15 1745 1594

1594 31 |Z-3| 1968
1968 21 1756 1922
1922 20 1745 1972
1972 60 1735 1710
1710 30 |Z-3| 1645
1645 15 1756 1923

(28)

(8003-2)=000...c₁.....c₂....
la mantisse recadrée et l'exposant corrigé, la suite est prise en 1923 à l'entrée du SP de SORTIE.

(27) 1822 69 1826 1629

On a $Z=3$, il faut donc décaler la mantisse vers la gauche pour la recadrer sur la position 7 de 8003.

(1594) = 35 000 |Z-3| 1674

(1710) = 35 000 |Z-3| 1645

(8002) =c₃....(1745)=(8002)= ...c₃.. 000

(1756)=(8003)= 0000

(8003-2)=0000 ..c₁.....c₂..(8003-2)=000...c₁.....c₂...00

(28)

(8003-2)=000...c₁.....c₂....
la mantisse recadrée et l'exposant corrigé, la suite est prise en 1923 à l'entrée du SP de SORTIE.

(24) 1644 45 1874 1623
(19)

En 1644, c₁ est nul. On teste c₂ qui est chargé en 8002.

1874 67 1700 1924
1924 16 1878 1990
1990 20 1700 1974

(1878) = 0000000010 +

(1700)=(8002)= 0000000(Y-10)

On diminue Y de 10 unités.

1974 65 1745 1926
1926 21 1745 1976
1976 10 1730 1735

(6)

(1745)=(8003)= 0000000000

(8003-2)=.....c₁.....c₂....
L'ensemble de la mantisse est décalé de 10 positions vers la gauche et on reprend le SP de RECADRAGE en 1735.

(19) 1623 67 1700 1928
 1928 16 1881 1891
 1891 20 1700 1729
 1729 60 1745 1829
 1829 44 1941 1858
 (29)
 1941 20 1745 1735 (6)

En 1623, c_1 et c_2 sont nuls.
 (1881) = 0000000020 +
 (1700) = (8002) = 0000000(Y-20)
 On diminue Y de 20 unités.
 (8003-2) = c_30000000000
 En 1858, la mantisse étant
 nulle, le résultat de l'opé-
 ration est nul et en charge
 1700 de séros.
 (1745) = (8002) = 0000000000

L'ensemble de la mantisse est
 décalé de 20 positions vers
 la gauche et on reprend le
 SP de RECHARGAGE en 1735.

7^e Sous-Programme de SORTIE

7.1 Test sur la position 8 de 8003.

A l'entrée, la mantisse recadrée est chargée, c_3 en 1745,
 les 2 autres tranches en 8003-2, les positions 10 et 9 de
 8003 étant libres mais la position 8 pouvant ne pas l'être.

(28) 1923 21 1715 1879 (1715) = (8003) = 003... c_1 ..
 1879 20 1730 1991 (1730) = (8002) = c_2
 1991 36 0000 1929 (8002) = xxxxxxx0Q

(8002) = xxxxxxx0(Q-2)

En 1563, nous avons Q=2. Re-
 marquons que dans ce cas les
 xxx..x sont des séros. La po-
 sition 8 de 8003 était donc
 occupée; il faut reprendre
 le recadrage de la mantisse.

7.2 Tests sur l'exposant

(25) 1889 65 1700 1630 (8002) = 0000000 Y
 1630 45 1692 1742
 (30)
 1692 46 1858 1646
 (29)

En 1742, Y est nul. La suite
 est prise pour charger le PA
 de séros puisque le résultat
 est nul.
 En 1858 Y est négatif. On
 considère donc le résultat
 comme nul et on charge le PA

de séros.

1646 16 1680 1792 (8002) = 0000000(Y-1000)
 1792 46 1795 1696
 01 0111 8000

En 1795, l'exposant est bien
 un nombre positif de 3 chiffres,
 c'est à dire tel que :

$$0.6 Y < 1000$$

7.3 Résultat sous forme "3 Mots"

1696 15 8001 1830 (8002) = 0000000YYY +
 1830 35 0007 1880
 1880 20 1587 1842 (1587) = (8002) = YYY 00000000 +
 1842 60 1715 1930 (8003-2) = 000... c_1 c_2
 1930 15 1730 1892 (8003-2) = YYY... c_1 c_2
 1892 46 1845 1796 (8003-2) = YYY... c_1 c_2
 1845 11 1587 1660 (8003-2) = YYY... c_1 c_2
 1796 10 1587 1660 (1746) = YYY... c_1 ..
 (21) 1660 21 1746 1651 (8001) = c_3
 1651 69 1745 1740

1740 Instruction
 de Sortie

En 1740 on retrouve l'instruction
 de sortie de l'Arithmétique.

A la sortie de l'Arithmétique le résultat de l'opération
 apparaît sous 2 formes :

Forme "3 Mots" : (8003-2) = YYY... c_1 c_2

(8001) = c_3

et (1746) = YYY... c_1 ..
 (1730) = c_2
 (1745) = c_3

Forme "4 Mots" : (1700) = 0000000 YYY +
 (1715) = 000... c_1 ..
 (1730) = c_2
 (1745) = c_3

ANNEXE 4

ARITHMETIQUE COMPLEXE SIMPLE PRECISION A. Cx. S. P. ou B.B.L. D pour Nombres Complexes, Virgule Flottante FLAIR Simple Précision

Entrées dans l'Arithmétique

1750 : Addition
1751 : Soustraction
1752 : Division
1753 : Multiplication
1754 : Transfert du P.A. vers la mémoire générale
1755 : Transfert dans le P.A.
1756 : Transfert dans le P.A. de l'opposé
1796 : Chargement de la distance entre partie réelle et partie imaginaire

Macro-Code Symbolique

CAA
CAS
CDV
CMU
CTR
CZA
CZS
CZZ

Sorties de l'Arithmétique

en 1740 dans le cas général ,
en 1677 dans le cas de l'opération CDV
en 8003 dans le cas des opérations CTR et CZZ

Pseudo-Accumulateur Complexe.

1650 $\alpha\alpha$a.... Partie réelle du premier opérande ou du résultat de l'opération.
 1655 $\beta\beta$b.... Partie imaginaire du premier opérande ou du résultat de l'opération
 Pseudo-Distributeur Complexe.
 1662 $\alpha\alpha'$a'.... Partie réelle du second opérande
 1667 $\beta\beta'$b'.... Partie imaginaire du second opérande.

Mémoires à adresse facteur variable

1658 60 F 1656 Appel de la partie réelle dans 8003
 1659 61 F 1656 Appel de l'opposé de la partie réelle dans 8003.
 1671 61 G 1653 Appel de l'opposé de la partie imaginaire dans 8003.
 1672 60 G 1654 Appel de la partie imaginaire dans 8003.
 1721 60 G 1653 Appel de la partie imaginaire dans 8003
 1768 60 F 1657 Appel de la partie réelle dans 8003

Mémoires de travail.

Ce sont les mémoires 1665, 1677, 1685 et 1762. Ces mémoires sont utilisées pour le transfert des instructions construites par le Sous-Programme et le transfert provisoire des résultats intermédiaires.

En outre :

1762 Orientation : Première instruction du SP Opérationnel sélectionné.
 1732 00 Δ 0000 Δ est la distance entre partie réelle et la partie imaginaire des nombres traités.
 1740 Inst.Sortie Contient l'instruction à prendre à la suite de l'opération en cours. Cette instruction de sortie se trouve généralement en 1677 à la sortie du SP de DIVISION.

Mémoire à contenu fixe

1691 01 0333 8000 La distance Δ est nulle ou bien n'a pas été chargée en 1732.

L'entrée dans l'Arithmétique est spécifique de chaque opération : on distingue 4 opérations arithmétiques, 2 transferts vers le PAC et 1 transfert vers la mémoire générale. A l'entrée dans l'Arithmétique, les contenus de 8003 et 8002 sont les suivants :

(8003) = Instruction de Sortie
 (8002) = 00 F xxxx
 F étant l'adresse de la partie réelle du nombre traité par l'opération

1750 69 1703 1709 On charge 8001 de la première instruction du SP Opérationnel correspondant à l'opération.
 1751 69 1704 1709
 1752 69 1705 1709
 1753 69 1706 1709

1709 24 1762 1715 (1762) = (8001)
 1715 69 1768 1771 (8001) = 60 xxxx 1657
 (8002) = 00 F xxxx
 1771 22 1665 1669 (1665) = 60 F 1657
 1669 69 1672 1674 (1) (8001) = 60 xxxx 1654

1756 69 1659 1712 (8001) = 61 xxxx 1656
 1712 22 1665 1668 (1665) = 61 F 1656
 1668 69 1671 1674 (1) (8001) = 61 xxxx 1653

1755 69 1658 1661 (8001) = 60 xxxx 1656
 1661 22 1665 1718 (1665) = 60 F 1656
 1718 69 1721 1674 (1) (8001) = 60 xxxx 1653

(1) 1674 22 1677 1680 (1677) = 6⁰₁ F 165³₄
 1680 21 1740 1693 (1740) = Instruction de Sortie
 1693 65 1732 1687 (8002) = 00 Δ 0000
 1687 45 1690 1691 En 1691, la distance Δ étant nulle ou n'ayant pas été chargée, le déroulement du programme est interrompu :

(1691) = 01 0333 8000
 (8002) = 6⁰₁ (F+ Δ) 165³₄
 (F+ Δ) = G, adresse de la partie imaginaire.

Cas des opérations arithmétiques

Nous avons alors : (8002) = 60 G 1654

8002 60 G 1654 (8003) = $\beta\alpha' \dots b' \dots$
 1654 21 1667 1665 (1667) = $\beta\beta' \dots b' \dots$
 1665 60 F 1657
 1657 21 1662 1762 (1662) = $\alpha\alpha' \dots a' \dots$
 En 1762 on retrouve la première instruction du SP Opérationnel sélectionnée lors de l'entrée.

REMARQUE

Chargement de la distance $\Delta\Delta\Delta\Delta$ en 1732.

Les contenus de 8003 et 8002 étant les suivants :

(8003) = Instruction de Sortie
 (8002) = 00 A 0000 +

l'instruction de chargement est :

1796 20 1732 8003 (1732) = 00 A 0000 +

2° Sous-Programmes de TRANSFERT

2.1 Transfert vers le PACx

2.1.1 Transfert du nombre complexe (A,B)

L'entrée dans le Sous-Programme d'Entrée est en 1755, de sorte que la sortie de ce Sous-Programme, les contenus de 1665 et 8002 sont les suivants :

(1665) = 60 F 1656 et (8002) = 60 G 1653

8002 60 G 1653
 1653 21 1655 1665 (1655) = $\beta\alpha' \dots b' \dots$
 1665 60 F 1656
 1656 21 1650 1740 (1650) = $\alpha\alpha' \dots a' \dots$
 1740 Inst.Sortie En 1740 on retrouve l'instruction de sortie de l'Arithmétique.

2.1.2 Transfert de l'opposé du nombre complexe (A,B)

L'entrée dans le Sous-Programme d'Entrée est en 1756, de

sorte que la sortie de ce Sous-Programme les contenus de 1665 et 8002 sont les suivants :

(1665) = 61 F 1656 et (8002) = 61 G 1653

8002 61 G 1653
 1653 21 1655 1665 (1655) = $-\beta\beta' \dots b' \dots$
 1665 61 F 1656
 1656 21 1650 1740 (1650) = $-\alpha\alpha' \dots a' \dots$
 1740 Inst.Sortie En 1740 on retrouve l'instruction de sortie de l'Arithmétique.

2.2 Transfert vers la mémoire générale

Le résultat de l'opération est chargé dans le PACx sous la forme :

(1650) = $\gamma\gamma' \dots c' \dots$ et (1655) = $\delta\delta' \dots d' \dots$

Il doit être transféré dans les mémoires générales d'adresses F et G (G=F+ Δ), F étant l'adresse indiquée en position d'adresse facteur de 8002.

A l'entrée : (8003) = Instruction de Sortie

(8002) = 00 F xxxx

1754 69 1742 1795 (8001) = 24 xxxx 1652
 1795 22 1677 1792 (1677) = 24 F 1652
 1792 15 1732 1793 (8002) = 00 (F+ Δ) xxxx
 (F+ Δ) = G
 1793 69 1696 1700 (8001) = 24 xxxx 8003
 1700 22 1665 1694 (1665) = 24 G 8003
 1694 69 1650 1677 (8001) = $\gamma\gamma' \dots c' \dots$
 1677 24 F 1652 Partie réelle du résultat transférée en F
 1652 69 1655 1665 (8001) = $\delta\delta' \dots d' \dots$
 1665 24 G 8003 Partie imaginaire du résultat transférée en G
 8003 Inst.Sortie En 8003 on retrouve l'instruction de Sortie de l'Arithmétique.

3° Sous-Programme d'ADDITION

A l'entrée dans le Sous-Programme d'ADDITION, le premier opérande (A,B) est déjà chargé dans le PACx et la second opérande (A',B') a été chargé dans le PDCx au cours du passage

3.1 ADDITION de 2 nombres complexes: (1762)=(1703)

(2) 1703 60 1662 1717
 1717 69 1650 1707
 1707 24 1801 1757
 1757 69 1902 1708
 1708 24 1808 1711
 1711 69 1664 1800

(8003) = ...A'....
 (1801) =A....
 (1808) = 980800889
 Code Addition pour AVF
 (8001) = 21 1650 1758

(8003) = ...C.... = (A+A')
 (1803) = (1664)
 (1650) =C....
 Partie réelle du résultat
 chargée en 1650
 (8003) = \B...
 (1801) = ...B'....
 (8001) = 21 1655 1740

1800
 A.V.F.
 1803

1664 21 1650 1758
 1758 60 1655 1758
 1758 69 1667 1670
 1670 24 1801 1660
 1660 69 1663 1800

1800
 A.V.F.
 1803

1663 21 1655 1740

1740 Inst.Sortie

(8003) =D.... = (B+B')
 (1803) = (1663)
 (1655) =D....
 Partie réelle du résultat
 chargée en 1655
 Le résultat de l'addition
 étant chargé dans le PACx, l'
 instruction de sortie de l'
 Arithmétique est prise en 1740

3.2 SOUSTRACTION de 2 nombres complexes: (1762)=(1704)

On change le signe de la partie réelle et de la partie
 imaginaire du second opérande avant de prendre la suite des
 instructions à l'entrée du Sous-Programme d'Addition.

1704 61 1667 1722
 1722 21 1667 1720
 1720 61 1662 1717 (2) (8003) = -(A'-B')
 (8003) = -(A'-B')

4° Sous-Programme de MULTIPLICATION

A l'entrée dans le Sous-Programme de MULTIPLICATION, le
 premier opérande (A,B) est déjà chargé dans le PACx et le se-
 cond opérande (A',B') a été chargé dans le PACx au cours du
 passage par le Sous-Programme d'Entrée.

4.1 Calcul de la partie réelle: (1762)=(1706)

(3) 1706 60 1650 1710
 1710 69 1662 1765
 1765 24 1801 1760
 1760 69 1996 1699
 1699 24 1808 1761
 1761 69 1714 1800

(8003) =A....
 (1801) =A'....
 (1808) = 0008900888
 Code multiplication pour l'AVF
 (8001) = 21 1685 1688

1800
 A.V.F.
 1803

1714 21 1685 1688
 1688 61 1667 1772
 1772 69 1655 1713
 1713 24 1801 1763
 1763 69 1666 1800

(8003) = AxA'
 (1803) = (1714)
 (1685) = AxA'
 (8003) = -(....B'....)
 (1801) =B....
 (8001) = 69 1685 1768

1800
 A.V.F.
 1803

1666 69 1685 1738
 1738 24 1801 1764
 1764 69 1902 1716
 1716 24 1808 1766
 1766 69 1719 1800

(8003) = -BxB'
 (1803) = (1666)
 (1801) = AxA'
 (1808) = 980800889
 Code Addition pour l'AVF
 (8001) = 21 1762 1767

1800
 A.V.F.
 1803

(8003) = AxA' - BxB'
 partie réelle du produit.
 (1803) = (1719)

1719 21 1762 1767

(1762) = $A \times A' - B \times B'$ =
 $Y Y' \dots c \dots$ =
partie réelle du produit.

4.2 Calcul de la partie imaginaire.

1767 60 1650 1769

(8003) = $\dots A \dots$

1769 69 1667 1770

(1801) = $\dots B' \dots$

1770 24 1801 1673

(1808) = 9008900888

1673 69 1996 1749

(8001) = 21 1685 1789

1749 24 1808 1723

1723 69 1676 1800

1800
A.V.F.
.. 1803

(8003) = $A \times B'$

(1803) = (1676)

(1685) = $A \times B'$

(8003) = $\dots B \dots$

1676 21 1685 1788

1788 60 1655 1773

1773 69 1662 1724

1724 24 1801 1774

1774 69 1727 1800

(1801) = $\dots A' \dots$

(8001) = 69 1685 1689

1800
A.V.F.
.. 1803

(8003) = $B \times A'$

(1803) = (1727)

(1801) = $A \times B'$

1727 69 1685 1689

1689 24 1801 1675

1675 69 1902 1725

1725 24 1808 1775

1775 69 1678 1800

(1808) = 9800800889

Code Addition pour l'AVF

(8001) = 21 1655 1726

1800
A.V.F.
.. 1803

(8003) = $B \times A' + A \times B'$

Partie imaginaire du produit

(1803) = (1678)

(1655) = $B \times A' + A \times B'$ =
 $\delta \delta \dots d \dots$

partie imaginaire du produit.

1678 21 1655 1726

1726 69 1762 1776

1776 24 1650 1740

(1650) = $A \times A' - B \times B'$ =
 $Y Y' \dots c \dots$

La partie réelle du produit
est chargée dans le PACX.

1740 Inst.Sortie

En 1740, on retrouve l'ins-
truction de sortie de l'AVF-
rithmétique.

Partie réelle et partie imaginaire du produit sont char-
gées dans le PACX.

{ (1650) = $Y Y' \dots c \dots$
(1655) = $\delta \delta \dots d \dots$

5° Sous-Programme de DIVISION

A l'entrée dans le Sous-Programme de DIVISION le premier
opérande (A,B) est déjà chargé dans le PACZ et le second opé-
rande a été chargé dans le PDCI lors du passage par le Sous-
Programme d'Entrées.

Et nous avons (1762) = (1705)

5.1 Calcul du dénominateur du quotient

1705 60 1662 1777

(8003) = $\dots A' \dots$

1777 69 1662 1728

1728 24 1801 1778

(1801) = $\dots A' \dots$

1778 69 1996 1799

1799 24 1808 1679

(1808) = 0008900888

Code multiplication pour AVF

(8001) = 21 1685 1739

1679 69 1662 1800

1800
A.V.F.
.. 1803

(8003) = $(A')^2$

(1803) = (1682)

(1685) = $(A')^2$

(8003) = $\dots B' \dots$

(1801) = $\dots B' \dots$

(8001) = 69 1685 1789

1682 21 1685 1739

1739 60 1667 1729

1729 69 1667 1779

1779 24 1801 1730

1730 69 1683 1800

(8003) = $B \times A' + A \times B'$

Partie imaginaire du produit

(1803) = (1678)

(1655) = $B \times A' + A \times B'$ =
 $\delta \delta \dots d \dots$

partie imaginaire du produit.

1706
Sous-Programme
de Multiplication
.. 1740

(3)

$$\begin{aligned}(1650) &= A \times A' + B \times B' \\ (1655) &= A' \times B - A \times B' \\ (1740) &= (1686)\end{aligned}$$

5.3 Calcul de la partie réelle du quotient

1686 60 1650 1734
1734 69 1665 1784
1784 24 1801 1785
1785 69 1900 1736
1736 24 1808 1786
1786 69 1790 1800

1800
A.V.P.
.. 1803

$$(8003) = \frac{A \times A' + B \times B'}{(A')^2 + (B')^2} =$$

....C.... = partie réelle du quotient.

$$(1803) = (1790)$$

$$(1650) = 77 \dots c \dots$$

5.4 Calcul de la partie imaginaire du quotient

1737 60 1655 1787
1787 69 1665 1791
1791 24 1801 1692
1692 69 1745 1800

1800
A.V.P.
.. 1803

$$(8003) = \frac{A' \times B - A \times B'}{(A')^2 + (B')^2} =$$

....D.... = partie imaginaire du quotient.

$$(1803) = (1745)$$

$$(1655) = 77 \dots d \dots$$

En 1677 on retrouve l'instruction de Sortie de l'Arithmétique.

1745 21 1655 1677
1677 Inst.Sortie

$$\begin{aligned}(8003) &= (B')^2 \\ (1803) &= (1683)\end{aligned}$$

$$(1801) = (A')^2$$

$$(1808) = 9800800889$$

Code addition pour AVF

$$(8001) = 21 \ 1665 \ 1782$$

$$(8003) = (A')^2 + (B')^2$$

$$(1803) = (1684)$$

$$(1665) = (A')^2 + (B')^2$$

Le dénominateur est transféré en 1665 où il sera représenté ultérieurement.

5.2 Utilisation du SP de Multiplication

Pour calculer la partie réelle et la partie imaginaire du quotient des 2 opérands, il nous faut faire le produit $(A, B) \times (A', -B')$.

Pour cela, après avoir convenablement modifié le second opérande et adapté la sortie du SP de multiplication, on prend la suite des instructions en 1706.

On teste le dénominateur qui servira de diviseur dans la dernière partie du Sous-Programme.

Si $(A')^2 + (B')^2 = 0$, la suite est prise en 1876 à l'interieur de l'AVF et qui contient l'arrêt :

01 0222 8000

$$(1677) = (8002) = \text{Inst. de Sortie}$$

$$(1740) = (8002) = 60 \ 1650 \ 1734$$

$$(1667) = (8002) = -(\dots B' \dots)$$

(3)

1800
A.V.P.
.. 1803

1683 69 1685 1789
1789 24 1801 1780
1780 69 1902 1731
1731 24 1808 1781
1781 69 1684 1800

1800
A.V.P.
.. 1803

1684 21 1665 1782

1782 44 1735 1876

1735 65 1740 1695
1695 20 1677 1733
1733 65 1686 1741
1741 20 1740 1743
1743 66 1667 1783
1783 20 1667 1706

Partie réelle et partie imaginaire du quotient des 2 opérantes (a,b) et (a',b') sont chargées dans le PACx sous la forme :

$$\begin{cases} (1650) = \gamma\gamma\dots c\dots \\ (1655) = \delta\delta\dots d\dots \end{cases}$$

ANNEXE 5

ARITHMETIQUE COMPLEXE DOUBLE PRECISION A. Cx. D.P. ou B.B.L. E pour Nombres Complexes, Virgule Flottante FLAIR Double Précision

Entrées dans l'Arithmétique

Macro-Code Symbolique

1768 : Addition	GAA
1769 : Soustraction	GAS
1770 : Division	GDV
1771 : Multiplication	GMU
1772 : Transfert du P.A. vers la mémoire générale	GTR
1773 : Transfert dans le P.A. '	GZA
1774 : Transfert dans le P.A. de l'opposé	GZS
1798 : Chargement de la distance entre partie réelle et partie imaginaire et de la distance entre les 2 Mots de chacune de ces parties	GZZ

Sorties de l'Arithmétique

en 1625 dans le cas général	
en 1584 dans le cas de l'opération	GDV
en 8003 dans le cas de l'opération	GTR

Pseudo-Accumulateur Complexe.

1650	αα...α ₁ ...	Mot 1 de la partie réelle du premier opérande ou du résultat de l'opération.
1665	...α ₂ ...	Mot 2 de la partie réelle du premier opérande ou du résultat de l'opération.
1655	ββ...β ₁ ...	Mot 1 de la partie imaginaire du premier opérande ou du résultat de l'opération.
1670	...β ₂ ...	Mot 2 de la partie imaginaire du premier opérande ou du résultat de l'opération.

Pseudo-Distributeur Complexe.

1662	α'α'...α' ₁ ...	Mot 1 de la partie réelle du second opérande
1677	...α' ₂ ...	Mot 2 de la partie réelle du second opérande.
1667	β'β'...β' ₁ ...	Mot 1 de la partie imaginaire du second opérande.
1682	...β' ₂ ...	Mot 2 de la partie imaginaire du second opérande.

Instructions à adresse facteur variable.

1548	24	G'	8003	Transfert du Mot 2 de la partie imaginaire du résultat dans la mémoire d'adresse G'.
1560	65	F'	1553	Appel du Mot 2 de la partie réelle du nombre traité.
1566	69	G'	1652	Appel du Mot 2 de la partie imaginaire du nombre traité.
1600	24	G	1664	Transfert du Mot 1 de la partie imaginaire du résultat dans la mémoire d'adresse G.
1624	10	F	1653	Appel du Mot 1 de la partie réelle du nombre traité.
1633	69	G	1552	Appel du Mot 1 de la partie imaginaire du nombre traité.
1690	24	F	1564	Transfert du Mot 1 de la partie réelle du résultat dans la mémoire d'adresse F.
1794	24	F'	1614	Transfert du Mot 2 de la partie réelle du résultat dans la mémoire d'adresse F'.

Mémoires de travail

Ce sont les mémoires 1602, 1603, 1604, 1617, 1618,

1619 et 1643.

Ces mémoires sont utilisées pour le transfert des instructions construites par le Programme et le transfert provisoire des résultats des calculs intermédiaires.

En outre :

1618	Orientation	Première instruction du SP Opérationnel sélectionné.
1581	00 (2DDD)' 0000	Distance entre Mot 1 et Mot 2 de la partie réelle et de la partie imaginaire des nombres traités.
1541	00 ΔΔΔΔ 0000	Distance entre les Mots 1 de la partie réelle et de la partie imaginaire et les Mots 2 de la partie réelle et de la partie imaginaire des nombres traités.
1625	Inst.Sortie	Contient l'instruction à prendre à la suite de l'opération en cours. Cette instruction de sortie se trouve en 1643 à la sortie du SP de DIVISION.

Mémoires à contenu fixe.

1568	00 0015 0000 +
1750	00 0100 0000 +
1595	00 1602 0000 +
1596	00 1604 0000 +
1626	00 1650 0000 +
1637	00 1662 0000 +
1674	00 1655 0000 +
1685	00 1667 0000 +
1751	00 1603 0000 +

1^o Sous-Programme d'ENTREE

L'entrée dans l'Arithmétique est spécifique de chaque opération : on distingue 4 opérations arithmétiques, 2 transferts vers le PAC et 1 transfert vers la mémoire générale. A l'entrée dans l'Arithmétique les contenus de 8003 et 8002 sont les suivants :

(8003) = Instruction de Sortie.

(8002) = 00 F xxxx
F étant l'adresse du Mot 1 de la
partie réelle du nombre traité par
l'opération.

1768 69 1571 1630
1769 69 1572 1630
1770 69 1573 1630
1771 69 1574 1630
1773 69 1576 1630
1774 69 1577 1630

On charge 8001 de la première
instruction du SP Opérationnel
correspondant à l'entrée, donc
à l'opération envisagée.

(1618) = (8001)
(8001) = 10 xxxx 1653
(8002) = 00 F xxxx
(1602) = 10 F 1653
(1625) = Instruction de
Sortie.

(8002) = 00 (DDDD)'0000

En 1741 la distance (DDDD)'est
nulle ou bien n'a pas été char-
gée en 1581. Le déroulement
du programme est interrompu

(1741) = 01 0353 8000

(8003) = 00 ΔΔΔΔ 0000

En 1741 la distance ΔΔΔΔ
est nulle ou bien n'a pas été
chargée en 1541. Voir ci-
dessus.

(8002) = 10 (F+D)' 1653
(F+D)' = F'
(8001) = 65 xxxx 1553
(1603) = 65 F' 1553
(8002) = 10 (F+D'+Δ) 1653
(F+D'+Δ) = G'
(8001) = 69 xxxx 1652
(1604) = 69 G' 1652
(8002) = 10 (F+Δ) 1653
(F+Δ) = G
(8001) = 69 xxxx 1552
(8001) = 69 G 1552

(8001) = 00'...b₁'....

(1667) = Mot 1 de la partie
imaginaire.

8001 69 G 1552
1552 24 1667 1604

(8001) =b₂'....
(1682) = Mot 2 de la partie
imaginaire.
(8002) =a₂'....
(1677) = Mot 2 de la partie
réelle.
(8003) = 00'...a₁'....
(1662) = Mot 1 de la partie
réelle.

Le PDCx est donc en totalité
chargé des différents Mots
du second opérande.

(8001) = 00 0015 0000 +
(1732) = 00 0015 0000 +
Comme l'Arithmétique Complexe
utilise l'ADP, on charge dans
la mémoire 1732 de cette der-
nière la distance séparant
les mémoires contenant les
Mot 1 et 2 des nombres que l'
ADP devra traiter.

En 1618 on retrouve la premi-
ère instruction du SP Opéra-
tionnel sélectionné à l'entrée.

Chargement des distances ΔΔΔΔ et (DDDD)' dans les mém-
res 1541 et 1581. A l'entrée les contenus de 8003 et 8002
sont les suivants :

(8003) = Instruction de Sortie
(8002) = 00 ΔΔΔΔ (DDDD)'

1798 21 1625 1796 (1625) = Instruction de
Sortie.
1796 60 8002 1616 (8003) = 00 ΔΔΔΔ (DDDD)'
1616 30 0004 1597 (8003) = 00 0000
(8002) = (DDDD)'00 0000
(8003-2)=00(DDDD)'0000ΔΔΔΔ00000000
(1581)=(8003) = 00 (DDDD)' 0000
(1541)=(8002) = 00 ΔΔΔΔ 0000
En 1625 on retrouve l'instruc-
tion de Sortie de l'Arithmé-
tique.

A l'entrée : (8003) = Instruction de Sortie
(8002) = 00 F xxxx

1772 69 1690 1694 (8001) = 24 xxxx 1564
1694 22 1602 1661 (1602) = 24 F 1564
1661 15 1581 1790 (8002) = 00 (F+D') xxxx
1790 69 1794 1547 (8001) = 24 xxxx 1614
1547 22 1603 1562 (1603) = 24 F' 1614
F' = (F+D')

1562 15 1541 1795 (8002) = 00 (F+D'+Δ) xxxx

1795 69 1548 1651 (8001) = 24 xxxx 8003

1651 22 1604 1612 (1604) = 24 G' 8003

G' = (F+D'+Δ)

1612 16 1581 1646 (8002) = 00 (F+Δ) xxxx

1646 69 1600 1663 (8001) = 24 xxxx 1664

1663 22 1617 1696 (1617) = 24 G 1664

G = (F+Δ)

1696 69 1650 1602 (8001) = 14...c₁...

1602 24 F 1564 Mot 1 de la partie réelle

du résultat transféré en F

1564 69 1665 1603 (8001) =c₂....

1603 24 F' 1614 Mot 2 de la partie réelle

du résultat transféré en F'

1614 69 1655 1617 (8001) = 16...d₁....

1617 24 0 1664 Mot 1 de la partie imaginaire

du résultat transféré en G

1664 69 1670 1604 (8001) =d₂....

1604 24 G' 8003 Mot 2 de la partie imaginaire

du résultat transféré en G'

8003 Inst.Sortie Les 4 Mots du résultat ayant

été transférés en mémoire générale, l'instruction de

Sortie est prise en 8003.

3° Sous-Programme d'ADDITION

A l'entrée dans le Sous-Programme d'ADDITION, le premier opérande (A,B) est déjà chargé dans le PACx et le second opérande (A',B') a été chargé dans le PDCx au cours du passage par le Sous-Programme d'Entrée.

3.1 ADDITION de 2 nombres complexes: (1618)-(1571)

2° Sous-Programmes de TRANSFERT

2.1 Transferts vers le PACx

A l'entrée la totalité du nombre a été chargée dans le PDCx, mais 8003 et 8002 contiennent encore les Mots 1 et 2 de la partie réelle.

2.1.1 Transfert du nombre complexe (A',B')

Nous avons (1618) = (1576)

1576 21 1650 1753 (1650)=(8003)= a'a'...a₁'...

1753 20 1665 1668 (1665)=(8002)=a'₂....

1668 60 1667 1622

1622 15 1682 1587 (8003-2)= a'a'...b'₁'.....b'₂'....

(1) 1587 21 1655 1558 (1655) = a'a'...b'₁'....

1558 20 1670 1625 (1670) =b'₂....

1625 Inst.Sortie La totalité du nombre ayant été chargée dans le PACx, la suite des instructions est prise en 1625 qui contient l'instruction de sortie de l'Addition.

2.1.2 Transfert de l'opposé du nombre complexe (A',B')

Nous avons (1618) = (1577)

1577 61 1662 1567

1567 16 1677 1631

(8003-2) = -(a'a'...a'₁'.....a'₂'....)

1631 21 1650 1534 (1650) = -(a'a'...a'₁'....)

1534 20 1665 1669 (1665) = -(....a'₂'....)

1669 61 1667 1672

1672 16 1682 1587 (1) (8003-2) = -(a'a'...b'₁'.....b'₂'....)

Transfert vers la mémoire générale

Le résultat de l'opération est chargé dans le PACx sous la forme:

(1650) = 14...c₁... (1665) =c₂....

(1655) = 16...d₁... (1670) =d₂....

Il doit être transféré dans les mémoires générales d'adresses F, F', G et G', F étant l'adresse indiquée en position d'adresse facteur de 8002.

(8002) = 00 1655 0000
(8003) = 65 1585 1589

(PA) = ... B ...
(1740) = (1582)

(8002) = 00 1667 0000
(8003) = 21 1655 1608

Addition de B'
(PA) = B + B' = (8003-2)
(1740) = (1542)

(1655) = (8003) : Mot 1 de la
partie imaginaire de la somme.
(1670) = (8002) : Mot 2 de la
partie imaginaire de la somme.
(8002) = 00 1650 0000
(8003) = 65 1637 1641

(PA) = ... A ...
(1740) = (1584)

(8002) = 00 1662 0000
(8003) = 21 1650 1654

Addition de A'
(PA) = A + A' = (8003-2)
(1740) = (1544)

(1650) = (8003) : Mot 1 de la
partie réelle de la somme
(1665) = (8002) : Mot 2 de la
partie réelle de la somme.
Le résultat de l'addition é-
tant chargé dans le PACx, la
suite est prise en 1625 où
l'on retrouve l'instruction
de sortie de l'Arithmétique.

3.2 SOUSTRACTION de 2 nombres complexes: (1618)=(1572)

On change le signe de la partie réelle et de la partie

1571 65 1674 1579
1579 10 1582 1764

1764
A.D.P.
.. 1740

1582 65 1585 1589
1589 10 1542 1757

1757
A.D.P.
.. 1740

1542 21 1655 1608
1608 20 1670 1623
1623 65 1626 1681
1681 10 1584 1764

1764
A.D.P.
.. 1740

1584 65 1637 1641
1641 10 1544 1757

1757
A.D.P.
.. 1740

1544 21 1650 1654

1654 20 1665 1625

1625 Inat.Sortie

imaginaire du second opérande avant de reprendre la suite
des instructions à l'entrée du Sous-Programme d'Addition.

1572 61 1662 1570
1570 16 1677 1632
1632 21 1662 1615
1615 20 1677 1580
1580 61 1667 1673
1673 16 1682 1687
1687 21 1667 1620
1620 20 1682 1571 (2)

Et c'est avec cet opérande
ainsi modifié que l'on re-
prend le Sous-Programme d'
Addition.

4° Sous-Programme de MULTIPLICATION

À l'entrée dans le Sous-Programme de MULTIPLICATION, le
premier opérande (A,B) est déjà chargé dans le PACx et le
second opérande (A',B') a été chargé dans le PDCx lors du
passage par le Sous-Programme d'Entrée.

Nous avons également : (1618) = (1574)

4.1 Calcul de la partie réelle du produit

(3) 1574 65 1626 1583 (8002) = 00 1650 0000
1583 10 1586 1764 (8003) = 65 1637 1691

1764
A.D.P.
.. 1740

(PA) = ... A ...
(1740) = (1586)

1586 65 1637 1691 (8002) = 00 1662 0000
1691 10 1594 1761 (8003) = 21 1602 1605

Multiplication par A'
(PA) = AxA' = (8003-2)
(1740) = (1594)

1594 21 1602 1605

1605 20 1617 1575
Transfert provisoire de $A \times A'$
en 1602 et 1617.

1575 65 1674 1629
(8002) = 00 1655 0000
(8003) = 65 1685 1639

1765
A.D.P.
.. 1740

(PA) = - B
(1740) = (1633)

1633 65 1685 1639
(8002) = 00 1667 0000
(8003) = 65 1595 1599

Multiplication par B'
(PA) = $-B \times B'$ = (8003-2)
(1740) = (1592)

1592 65 1595 1599
(8002) = 00 1602 0000
(8003) = 21 1603 1606

Addition de $A \times A'$
(PA) = $A \times A' - B \times B'$ = (8003-2)
Partie réelle du produit.
(1740) = (1752)

1752 21 1603 1606
Transfert provisoire de la
partie réelle du produit dans
1603 et 1618.

1757
A.D.P.
.. 1740

4.2 Calcul de la partie imaginaire du produit.

1675 65 1626 1683
(8002) = 00 1650 0000
(8003) = 65 1685 1689

1764
A.D.P.
.. 1740

(PA) = .. A ..
(1740) = (1636)

1636 65 1685 1689
(8002) = 00 1667 0000
(8003) = 21 1602 1755

Multiplication par B'
(PA) = $A \times B'$ = (8003-2)
(1740) = (1642)

1761
A.D.P.
.. 1740

1642 21 1602 1755
Transfert provisoire de $A \times B'$
en 1602 et 1617.
(8002) = 00 1655 0000
(8003) = 65 1637 1791

1764
A.D.P.
.. 1740

(PA) = .. A' ..
(1740) = (1634)

1634 65 1637 1791
(8002) = 00 1682 0000
(8003) = 65 1595 1649

Multiplication par B
(PA) = $A' \times B$ = (8003-2)
(1740) = (1644)

1644 65 1595 1649
(8002) = 00 1602 0000
(8003) = 21 1655 1658

Addition de $A \times B'$
(PA) = $A' \times B + A \times B'$ = (8003-2)
Partie imaginaire du produit.
(1740) = (1754)

1754 21 1655 1658
Transfert de la partie imagi-
naire du produit dans les mé-
moires 1655 et 1670 du PACI.

1757
A.D.P.
.. 1740

(8003-2) = $A \times A' - B \times B'$ = par-
tie réelle du produit.

Transfert de la partie réel-
le du produit dans les mémoi-
res 1650 et 1665 du PACI.

Le produit calculé ayant été
en totalité chargé dans le
PACI, la suite est prise en
1625 où l'on retrouve l'ins-
truction de Sortie de l'Arith-
métique.

1628 60 1603 1657
1657 15 1618 1544

1544 21 1650 1654
1654 20 1665 1625

1625 Inst.Sortie

5° Sous-Programme de DIVISION

A l'entrée dans le Sous-Programme de DIVISION le premier opérande, (A,B) est déjà chargé dans le PACx et le second opérande a été chargé dans le PDCx lors du passage par le Sous-Programme d'Entrée.

Et nous avons (1618) = (1573)

5.1 Calcul du dénominateur du quotient

1573 65 1637 1692 (8002) = 00 1662 0000
1692 10 1645 1764 (8003) = 65 1637 1792

1764
A.D.P.
.. 1740

(PA) = ..A'..
(1740) = (1645)

1645 65 1637 1792 (8002) = 00 1662 0000
1792 10 1695 1761 (8003) = 21 1602 1656

1761
A.D.P.
.. 1740

Multiplication par A'
(PA) = (A')² = (8003-2)
(1740) = (1695)

Transfert provisoire de A'²
en 1602 et 1617.

(8002) = 00 1667 0000
(8003) = 65 1685 1540

(PA) = ..B'..
(1740) = (1543)

(8002) = 00 1667 0000
(8003) = 65 1595 1699

Multiplication par B'
(PA) = (B')²
(1740) = (1593)

(8002) = 00 1602 0000
(8003) = 45 1610 1779

1764
A.D.P.
.. 1740

1543 65 1685 1540
1540 10 1593 1761

1761
A.D.P.
.. 1740

1593 65 1595 1699
1699 10 1756 1757

1757
A.D.P.
.. 1740

1756 45 1610 1779

1610 21 1604 1559
1559 20 1619 1680

5.2 Utilisation du SP de Multiplication

Pour calculer la partie réelle et la partie imaginaire du quotient des 2 opérandes, il faut effectuer le produit

$$(A,B) \times (A',-B')$$

Pour cela, après avoir convenablement modifié le second opérande et adapté la sortie du SP de Multiplication, on prend la suite des instructions en 1574.

1680 61 1667 1684 (8003-2) = -B'
1684 16 1682 1787 (1667) = -(A'..B'1...)
1787 21 1667 1686 (1682) = -(....B'2.....)
1686 20 1682 1786 (1643) = Inst.Sortie
1786 69 1625 1688
1688 24 1643 1546
1546 69 1550 1609
1609 24 1625 1574 (3) (1625)=(1550)= 65 1626 1788

(3)
1574
Sous-Programme
de Multiplication
.. 1625

1650 et 1665 contiennent les 2
Mots de la quantité :

$$A \times A' + B \times B'$$

1655 et 1670 contiennent les 2
Mots de la quantité :

$$A' \times B - A \times B'$$

(1625) = (1550)

5.3 Calcul de la partie réelle du quotient

1750 65 1626 1788
1788 10 1693 1764

1764
A.D.P.
.. 1740

1693 65 1596 1551
1551 10 1659 1760

1760
A.D.P.
.. 1740

1659 21 1650 1660
1660 20 1665 1590

1590 65 1674 1640
1640 10 1793 1764

1764
A.D.P.
.. 1740

1793 65 1596 1601
1601 10 1561 1760

1760
A.D.P.
.. 1740

1561 21 1655 1611
1611 20 1670 1643

(8002) = 00 1650 0000
(8003) = 65 1596 1551

(PA) = $A \times A' + B \times B'$
(1740) = (1693)

(8002) = 00 1604 0000
(8003) = 21 1650 1660

Division par le dénominateur
(PA) = (8003-2) = .. 0 ..

= $\frac{A \times A' + B \times B'}{(A')^2 + (B')^2}$ = partie réelle du quotient.
(1740) = (1659)

(1650) = Mot 1 de la partie réelle du quotient.
(1665) = Mot 2 de la partie réelle du quotient.

5.4 Calcul de la partie imaginaire du quotient.

(8002) = 00 1655 0000
(8003) = 65 1596 1601

(PA) = $A' \times B - A \times B'$
(1740) = (1793)

(8002) = 00 1604 0000
(8003) = 21 1655 1611

Division par le dénominateur
(PA) = (8003-2) = ... D ...
= $\frac{A' \times B - A \times B'}{(A')^2 + (B')^2}$ = partie imaginaire du quotient.

(1740) = (1561)
(1655) = Mot 1 de la partie imaginaire du quotient
(1670) = Mot 2 de la partie imaginaire du quotient.

1643 Inet,Sortie

En 1643 on retrouve l'instruction de Sortie de l'Arithmétique.

Partie réelle et partie imaginaire du quotient des 2 operands (A,B) et (A',B') sont chargées dans le PACx sous la forme :

(1650) = Mot 1 partie réelle
(1665) = Mot 2 partie réelle
(1655) = Mot 1 partie imaginaire
(1670) = Mot 2 partie imaginaire

ANNEXE 6

ARITHMETIQUE COMPLEXE MIXTE DOUBLE SIMPLE PRECISION

A. Cx. SD. P ou B.B.L. F

pour Nombres Complexes, Virgule Flottante FLAIR

Simple ou Double Précision

Entrées dans l'Arithmétique

En Simple Précision , voir Annexe 4

En Double Précision, voir Annexe 5

Sorties de l'Arithmétique

en 1625 dans le cas général

en 1531 dans le cas des opérations

en 8003 dans le cas des opérations

CDV et GDV

CTR et GTR

CZZ et GZZ

Pseudo-Accumulateur Complexe.

1550	$\alpha\alpha\dots\alpha_1\dots$	Mot 1 de la partie réelle du premier opérande ou du résultat de l'opération.
1565	$\dots\alpha_2\dots$	Mot 2 de la partie réelle du premier opérande ou du résultat de l'opération. En Simple Précision, α_2 est nul.
1555	$\beta\beta\dots\beta_1\dots$	Mot 1 de la partie imaginaire du premier opérande ou du résultat de l'opération.
1570	$\dots\beta_2\dots$	Mot 2 de la partie imaginaire du premier opérande ou du résultat de l'opération. En Simple Précision β_2 est nul.

Pseudo-Distributeur Complexe

1562	$\alpha'\alpha'\dots\alpha'_1\dots$	Mot 1 de la partie réelle du second opérande.
1577	$\dots\alpha'_2\dots$	Mot 2 de la partie réelle du second opérande. En Simple Précision, α'_2 est nul.
1567	$\beta'\beta'\dots\beta'_1\dots$	Mot 1 de la partie imaginaire du second opérande.
1582	$\dots\beta'_2\dots$	Mot 2 de la partie imaginaire du second opérande. En Simple Précision, β'_2 est nul.

Instructions à adresse facteur variable.

1401	24	F'	1497	Transfert du Mot 2 de la partie réelle du résultat dans la mémoire d'adresse F'.
1410	69	G	1502	Appel du Mot 1 de la partie imaginaire du nombre traité.
1425	69	F	1552	Appel du Mot 1 de la partie réelle du nombre traité.
1430	69	G'	1452	Appel du Mot 2 de la partie imaginaire du nombre traité.
1460	24	F	8003	Transfert du Mot 1 de la partie réelle du résultat de l'opération dans la mémoire d'adresse F.
1488	69	F'	1402	Appel du Mot 2 de la partie réelle du nombre traité.
1500	24	G'	1554	Transfert du Mot 2 de la partie imaginaire du résultat de l'opération dans la mémoire d'adresse G'.
1599	24	G	1613	Transfert du Mot 1 de la partie imaginaire du résultat de l'opération dans la mémoire d'adresse G.

Mémoires de travail

Ce sont les mémoires 1602, 1603, 1604
1617, 1618, 1619 et 1531.

Ces mémoires sont utilisées pour le transfert provisoire des instructions contruites par le programme et le transfert provisoire des résultats intermédiaires.

En entre:

1619	Orientation	Première instruction du SP Opération- nel sélectionné.
1431	00 Δ 0000	Distance entre le Mot 1 de la partie réelle et le Mot 1 de la partie ima- ginaire d'une part, et entre le Mot 2 de la partie réelle et le Mot 2 de la partie imaginaire, d'autre part, des nombres traités.
1441	00 D' 0000	Distance entre le Mot 1 et le Mot 2 de la partie réelle et également dis- tance entre le Mot 1 et le Mot 2 de la partie imaginaire des nombres trai- tés.
		En Simple Précision, comme les Mots 2 n'existent pas, on doit avoir: (1441) = 00 0000 0000 +
1625	Inst.Sortie	Contient l'Instruction de Sortie à Prendre à la suite de l'opération en cours. A la sortie du SP de DIVISION cette instruction deSortie se trouve en 1531.

Mémoires à contenu fixe.

1451	00 0100 0000 +
1538	00 0015 0000 +
1461	00 1550 0000 +
1506	00 1555 0000 +
1517	00 1567 0000 +
1551	00 1603 0000 +
1574	00 1604 0000 +
1578	00 1602 0000 +
1621	00 1562 0000 +

L'entrée dans l'Arithmétique est spécifique de chaque opération. On distingue, tant en Simple Précision qu'en Double Précision :

- 4 opérations arithmétiques
- 2 transferts vers le PACx
- 1 transfert vers la mémoire Générale.

A l'entrée les contenus de 8003 et 8002 sont les suivants :

(8003) = Instruction de Sortie
(8002) = 00 F xxx

F étant l'adresse du Mot 1 de la partie réelle du nombre traité par l'opération.

1.1 Instructions communes aux opérations en Simple et en Double Précision.

1750 69 1403 1631

On charge 8001 de la première instruction du SP Opérationnel Simple Précision correspondant à l'entrée, donc à l'opération envisagée.

1751 69 1404 1631

1752 69 1405 1631

1753 69 1406 1631

1768 69 1421 1631

On charge 8001 de la première instruction du SP Opérationnel Double Précision correspondant à l'entrée, donc à l'opération envisagée.

1769 69 1422 1631

1770 69 1423 1631

1771 69 1424 1631

1755 69 1426 1631

On charge 8001 de la première instruction du SP de Transfert correspondant à l'entrée, donc à l'opération envisagée. Le Sous-Programme est le même, en Simple et en Double Précision.

1756 69 1427 1631

1773 69 1426 1631

1774 69 1427 1631

1631 24 1619 1472

(1619) = Première Instruction sélectionnée.

(8001) = 69 xxx 1552

(1602) = 69 F 1552

(1625) = Instruction de Sortie

(8002) = 00 A 0000

En 1691, la distance Δ étant nulle ou bien n'ayant pas été chargée, le déroulement du programme est interrompu :

(1691) = 01 0333 8000

(8003) = 00 D' 0000 +

1438 10 1441 1445

1445 44 1449 1400

1.2 Chargement de l'opérande (A,B) dans le PDCx, en Double Précision.

(1) 1449 15 1602 1407

Puisque D' n'est pas nulle l'opération actuelle doit être effectuée en Double Précision et la suite est prise en 1449.

(8002) = 69 F+A 1552

(8001) = 69 xxx 1502

(1617) = 69 G 1502 (G=F+A)

(8002) = 69(F+A+D')1502

(8001) = 69 xxx 1452

(1603) = 69 G' 1452

G' = F + A + D'

(8002) = 69 F+D' 1502

(8001) = 69 xxx 1402

(8001) = 69 F' 1402 (F'=F+D')

(1577) = Mot 2 de la partie réelle = ...a₂....

(1582) = Mot 2 de la partie imaginaire = ...b₂....

(1732) = 00 0015 0000 +

Comme l'Arithmétique Complexe utilise l'ASDP, on charge en 1732 dans cette dernière, la distance séparant les mémoires contenant les Mots 1 et 2 des nombres que l'ASDP devra traiter.

(1567) = Mot 1 de la partie imaginaire = (a₁...b₁)...

(1562) = Mot 1 de la partie réelle = (a₁...b₁)...

En 1619 on retrouve la première instruction du SP Opérationnel sélectionné.

1617 69 G 1502

1502 24 1567 1602

1602 69 F 1552

1552 24 1562 1619

1619 Première Instruction

1.3 Chargement de l'opérande (A,B) dans le PDCx, en Simple Précision.

Les contenus de 8003 et 8002 sont donc les suivants:

(8003) = 00 0000 0000

(8002) = 00 0000 0000

Le fait que D' soit nul signifie que l'opération doit être effectuée en Simple Précision et la suite est prise en 1400.

(2) 1400 24 1577 1480 (1577) = 0000000000 +
 1480 24 1582 1585 (1582) = 0000000000 +
 1585 15 1602 1457 (8002) = 69 P+A 1552
 1457 69 1410 1463 (8001) = 69 xxxx 1502
 1463 22 1617 8001 (8001) = 69 G 1502
 8001 69 G 1502 (3)

REMARQUE 1

Chargement des distances A et D' dans les mémoires 1431 et 1441. A l'entrée les contenus de 8003 et 8002 sont les suivants:

(8003) = Instruction de Sortie

(8002) = 00 A D'

1798 21 1625 1898 (1625) = Instruction de Sortie
 1898 60 8002 1948 (8003) = 00 AAAA DDD'D'
 1948 30 0004 1998 (8003) = 00 0000 AAAA
 1998 19 1451 1649 (8002) = D'D D'D 0000
 1649 30 0002 1600 (8003) = D'D D'D 0000
 1600 21 1441 1501 (1441) = (8003) = 00 DDD'D 0000
 1501 20 1431 1625 (1431) = (8002) = 00 AAAA 0000

1625 Inst.Sortie
 En 1625 on retrouve l'Instruction de Sortie de l'Arithmétique.

REMARQUE 2

Chargement de la distance A en 1431 et de séries en 1441 dans le cas d'une utilisation de l'Arithmétique en Simple Précision. Les contenus de 8003 et 8002 sont les suivants:

(8003) = Instruction de Sortie

(8002) = 00 AAAA 0000

1796 20 1431 1648 (1431) = 00 AAAA 0000 +
 1648 60 8003 1614 (8003) = Inst.Sortie
 1614 20 1441 8003 (1441) = (8002) = 0000000000 +
 8003 Inst.Sortie

2° Sous-Programmes de TRANSFERT

2.1 Transferts vers le PACr

A l'entrée dans le Sous-Programme la totalité de l'opérande a été chargée dans le PDCr. Les 2 Sous-Programmes de Transferts vers le PACr sont communs à la Simple et à la Double Précision. Dans le cas de la Simple, les mémoires 1565 et 1570 seront chargées de séries.

2.1.1 Transfert du nombre complexe (A,B)

Nous avons (1619) = (1426)

1426 60 1567 1471 (8003-2) = partie imaginaire
 1471 15 1582 1437 (1555) = Mot 1 partie imaginaire
 1437 21 1555 1408 (1570) = Mot 2 partie imaginaire.
 1408 20 1570 1473
 1473 60 1562 1417
 1417 15 1577 1481 (8003-2) = partie réelle
 1461 20 1565 1418 (1565) = Mot 2 partie réelle
 1418 21 1550 1625 (1550) = Mot 1 partie réelle.

1625 Inst.Sortie
 La totalité du nombre ayant été chargée dans le PACr, avec (1570) = (1565) = 0000000000 s'il s'agit d'un nombre complexe Simple Précision, on retrouve en 1625 l'Instruction de Sortie de l'Arithmétique.

2.1.2 Transfert de l'opposé du nombre complexe (A,B)

Nous avons (1619) = (1427)

1427 61 1567 1521 (8003-2) = opposé de la partie imaginaire
 1521 16 1582 1487 (1555) = Opposé du Mot 1 de la partie imaginaire.
 1487 21 1555 1458 (1570) = Opposé du Mot 2 de la partie imaginaire.
 1458 20 1570 1523
 1523 61 1562 1467
 1467 16 1577 1481 (4) (8003-2) = Opposé de la partie réelle.

2.2 Transferts vers la mémoire générale.

2.21 Transfert en Double Précision.

Le résultat de l'opération étant chargé dans le PACX sous le forme :

$$\begin{aligned} (1550) &= y \gamma \dots \alpha_1 \dots & (1565) &= \dots \alpha_2 \dots \\ (1555) &= \delta \delta \dots \alpha_1 \dots & (1570) &= \dots \alpha_2 \dots \end{aligned}$$

doit être transféré dans les mémoires générales d'adresses F, F', G et G', F étant l'adresse indiquée en position d'adresse facteur en 8002.

A l'entrée : (8003) = Instruction de Sortie
(8002) = 00 F XIII

1772	69	1460	1563	(8001) = 24	xxxx	8003	
1563	22	1602	1560	(1602) = 24	F	8003	
1560	15	1431	1547	(8002) = 00	F+A	xxxx	(F+A=G)
1547	69	1599	1553	(9001) = 24	xxxx	1613	
1553	22	1617	1597	(1617) = 24	G	1613	
1597	15	1441	1647	(8002) = 00	F+A+D'	xxxx	
					G' = F+A+D'		
1647	69	1500	1454	(8001) = 24	xxxx	1554	
1454	22	1603	1610	(1603) = 24	G'	1554	
1610	16	1431	1548	(8002) = 00	F+D'	xxxx	(F+D=G')
1548	69	1401	1504	(8001) = 24	xxxx	1497	
1504	22	1618	1598	(1618) = 24	F'	1497	

1598	69	1570	1603	Mot 2 de la partie imaginai-
1603	24	01	1552	re transféré en G'

1554	69	1565	1618	Not 2 de la partie réelle transféré en P'
------	----	------	------	--

(5)	1497	69	1555	1617	Mot I de la partie imaginaire transféré en Q.
	1617	24	1517	1617	

1613	69	1550	1602	Not I de la partie réelle transféré en F.
------	----	------	------	---

8003 Inst.Sortie Les 4 Mots du résultat ayant été transférés en mémoire générale, l'Instruction de Sortie de l'Arithmétique est prise en 8003.

2.22 Transfert en Simple précision.

Le résultat de l'opération étant chargé dans le PACX
sous la forme :

$$\begin{array}{rcl} (1550) & = & r\gamma \dots \alpha_1 \dots \\ (1555) & = & s\delta \dots d_1 \dots \end{array} \quad \begin{array}{rcl} (1565) & = & 0000000000 \\ (1570) & = & 0000000000 \end{array}$$

doit être transféré dans les mémoires générales d'adresses P et Q, Y étant l'adresse indiquée en position d'adresse facteur de 8002.

A l'entrée : (8003) = Instruction de Sortie
(8002) = 00 F. XXX

1754	59	1460	1513	(8001) = 24	xxxx	8003
1533	22	1502	1510	(1602) = 24	F	8003
1510	15	1431	1646	(8002) = 00	T+A	xxxx
1646	69	1599	1602	(8001) = 24	xxxx	1613
1902	22	1517	1497	(5)	(1617) = 24	G 163.

3° Sous-Programmes d'ADDITION

A l'entrée dans le Sous-Programme d'ADDITION, en Simple ou en Double Précision, le premier opérande (A,B) est déjà chargé dans le PAX et le second opérande (A',B') a été chargé dans le PDCx au cours du passage par le Sous-Programme d'Entrée.

3.1 ADDITION de 2 nombres complexes

(6)	1403	65	1506	1411	(8002)	=	00	1555	0000
	1411	10	1414	1792	(8003)	=	65	1517	1571

(PA) = . . B ; ;
(1740) = (1414)

1414	65	1517	1571	(8002)	= 00	1567	0000
1571	10	1474	1786	(8003)	= 21	1555	1508

1786		Addition de B'
	A.S.D.P	$(PA) = \cdot B + R'$
	 1740	$(1740) = (1474)$

1474	21	1555	1508	(1555) = (8003) : partie imaginaire de la somme.
1508	65	1461	1415	(8002) = 00 1550 0000
1415	10	1468	1792	(8003) = 65 1621 1475

1792
 A.S.D.P.
 1740

(PA) = ..A..
 (1740) = (1468)

1468 65 1621 1475
1475 10 1418 1786

1786
A.S.D.P.
.. 1740

Addition de A'
(PA) = A + A'
(1740) = (1418)

1418 21 1550 1625

(1550) = (8003) = Partie imaginaire de la somme.

1625 Inst.Sortie

Le résultat de l'addition 6-tant chargé dans le PACz, l'Instruction de Sortie de l'Arithmétique est prise en 1625.

Remarquons que l'on a :

(1565)=(1570) = 0000000000.

3.2 SOUSTRACTION de 2 nombres complexes Simple Précision: (1619) = (1404)

On change le signe de la partie réelle et de la partie imaginaire du second opérande avant de reprendre la suite des instructions à l'entrée du SP d'Addition.

1404 61 1562 1518
1518 21 1562 1465
1465 61 1567 1522
1522 21 1567 1403 (6)

(1562) = Opposé de la partie réelle du second opérande.

(1567) = Opposé de la partie imaginaire du second opérande.

3.3 ADDITION de 2 nombres complexes Double Précision: (1619)=(1421)

(7) 1421 65 1506 1412
1412 10 1466 1764

1764
A.S.D.P.
.. 1740

(PA) = ..B..
(1740) = (1466)

1466 65 1517 1624
1624 10 1530 1757

(8002) = 00 1567 0000
(8003) = 21 1555 1608

1757
A.S.D.P.
.. 1740

Addition de B'
(PA) = B + B'
(1740) = (1530)

(1555) = (8003) = Mot 1 de la partie imaginaire de la somme.
(1570) = (8002) = Mot 2 de la partie imaginaire de la somme.
(8002) = 00 1550 0000
(8003) = 65 1621 1630

(PA) = .. A ..
(1740) = (1519)

(8002) = 00 1562 0000
(8003) = 20 1565 1418

Addition de A'
(PA) = A + A'
(1740) = (1481)

(1565) = (8002) = Mot 2 de la partie réelle de la somme
(1550) = (8003) = Mot 1 de la partie réelle de la somme.

Le résultat de l'addition étant chargé dans le PACz, l'Instruction de Sortie de l'Arithmétique est prise en 1625.

3.4 SOUSTRACTION de 2 nombres complexes Double Précision: (1619) = (1422)

On change le signe de la partie réelle et de la partie imaginaire du second opérande avant de reprendre la suite des instructions à l'entrée du SP d'Addition.

1422 61 1562 1569
1569 16 1577 1581
1581 21 1562 1566

(1562) = Opposé du Mot 1 de la partie réelle du second opérande.

(1577) = Opposé du Mot 2 de la partie réelle du second opérande.

1532 61 1567 1632
1632 16 1582 1587
1587 21 1567 1520

(1567) = Opposé du Mot 1 de la partie imaginaire du second opérande.

1520 20 1582 1421 (7) (1582) = Opposé du Mot 2 de la partie imaginaire du second opérande.

4° Sous-Programme de MULTIPLICATION

A l'entrée dans le Sous-Programme de MULTIPLICATION, tant en Simple qu'en Double Précision, le premier opérande (A,B) est déjà chargé dans le PACr et le second opérande (A',B') a été chargé dans le PDCr lors du passage par le Sous-Programme d'Entrée.

4.1 MULTIPLICATION en Simple Précision

Nous avons alors : (1619) = (1406)

4.1.1 Calcul de la partie réelle du produit.

(8) 1406 65 1461 1515 (8002) = 00 1550 0000
1515 10 1568 1792 (8003) = 65 1621 1525

1792
A.S.D.P.
.. 1740

(PA) = .. A ..
(1740) = (1568)

1568 65 1621 1525 (8002) = 00 1562 0000
1525 10 1528 1790 (8003) = 21 1602 1505

1790
A.S.D.P.
.. 1740

Multiplication par A'
(PA) = Ax A'
(1740) = (1528)

1528 21 1602 1505 Transfert provisoire de Ax A' en 1602.

1505 65 1506 1511 (8002) = 00 1555 0000
1511 10 1464 1793 (8003) = 65 1517 1572

1793
A.S.D.P.
.. 1740

(PA) = - B
(1740) = (1464)

1464 65 1517 1572 (8002) = 00 1567 0000
1572 10 1575 1790 (8003) = 65 1578 1483

1790
A.S.D.P.
.. 1740

Multiplication par B'
(PA) = -BxB'
(1740) = (1575)

1575 65 1578 1483 (8002) = 00 1602 0000
1483 10 1436 1786 (8003) = 21 1603 1556

1786
A.S.D.P.
.. 1740

Addition de Ax A'
(PA) = Ax A' + BxB'
(1740) = (1436)

1436 21 1603 1556 (1603) = Ax A' + BxB'. Transfert provisoire de la partie réelle du produit en 1603.

4.1.2 Calcul de la partie imaginaire du produit

1556 65 1461 1615 (8002) = 00 1550 0000
1615 10 1419 1792 (8003) = 65 1517 1622

1792
A.S.D.P.
.. 1740

(PA) = .. A ..
(1740) = (1419)

1419 65 1517 1622 (8002) = 00 1567 0000
1622 10 1476 1790 (8003) = 21 1602 1605

1790
A.S.D.P.
.. 1740

Multiplication par B'
(PA) = Ax B'
(1740) = (1476)

1476 21 1602 1605 Transfert provisoire de Ax B' en 1602.

1605 65 1506 1561 (8002) = 00 1555 0000
1561 10 1514 1792 (8003) = 65 1621 1526

1792
A.S.D.P.
.. 1740

(PA) = .. A' ..
(1740) = (1514)

1514 65 1621 1526 (8002) = 00 1562 0000

1526 10 1429 1790
(8003) = 65 1578 1533

Multiplication par B
(PA) = A'xB
(1740) = (1429)

1429 65 1578 1533
1533 10 1486 1786

1786
A.S.D.P.
.. 1740

Addition de AxB'
(PA) = A'xB + AxB'
(1740) = (1486)

1486 21 1555 1558
(1555) = A'xB + AxB' =
partie imaginaire du produit.

1558 60 1603 1418
1418 21 1550 1625

La partie réelle du produit
est transférée en 1550 dans
le PACx.

1625 Inst.Sortie
Le produit calculé ayant été
chargé dans le PACx, la suite
est prise en 1625 où se trou-
ve l'instruction de Sortie de
l'Arithmétique.

Remarquons que :
(1565)=(1570) = 0000000000

4.2 MULTIPLICATION en Double Précision

Nous avons alors : (1619) = (1424)

4.2.1 Calcul de la partie réelle du produit.

(9) 1424 65 1461 1616
1616 10 1620 1764

1764
A.S.D.P.
.. 1740

(PA) = .. A ..
(1740) = (1620)

1620 65 1621 1633
1633 10 1586 1761

1761
A.S.D.P.
.. 1740

1586 21 1602 1557
1557 20 1617 1484

1484 65 1506 1462
1462 10 1534 1765

1765
A.S.D.P.
.. 1740

1534 65 1517 1584
1584 10 1637 1761

1761
A.S.D.P.
.. 1740

1637 65 1578 1634
1634 10 1588 1757

1757
A.S.D.P.
.. 1740

1588 21 1603 1607
1607 20 1618 1635

Transfert provisoire de la
partie réelle du produit en
1603 et 1618.

4.2.2 Calcul de la partie imaginaire du produit.

1635 65 1461 1636
1636 10 1489 1764

1764
A.S.D.P.
.. 1740

(PA) = .. A ..
(1740) = (1489)

1489 65 1517 1638
(8002) = 00 1550 0000

(8003) = 65 1517 1638

A.S.D.P.
.. 1740

(PA) = AxB'
(1740) = (1591)

1591 21 1602 1409
1409 20 1617 1539
1539 65 1506 1512
1512 10 1589 1764

Transfert provisoire de AxB'
en 1602 et 1617.
(8002) = 00 1555 0000
(8003) = 65 1621 1639

1764
A.S.D.P.
.. 1740

(PA) = ..A'..
(1740) = (1589)

1589 65 1621 1639
1639 10 1442 1761

(8002) = 00 1562 0000
(8003) = 65 1578 1540

1761
A.S.D.P.
.. 1740

Multiplication par B
(PA) = A'xB
(1740) = (1442)

1442 65 1578 1540
1540 10 1443 1757

(8002) = 00 1602 0000
(8003) = 21 1555 1459

1757
A.S.D.P.
.. 1740

Addition de AxB'
(PA) = A'xB + AxB' = partie
imaginaire du produit.
(1740) = (1443)

1443 21 1555 1459
1459 20 1570 1590

Transfert de la partie imagi-
naire du produit dans les mé-
moires 1555 et 1570 du PACx.

1590 60 1603 1509
1515 15 1618 1481
1511 20 1565 1418
1418 21 1550 1625

(8003-2) = AXA' - BxB'

La partie réelle du produit
est transférée dans les mé-
moires 1550 et 1565 du PACx.

1625 Inst.Sortie

Le produit calculé ayant été
chargé dans le PACx, on re-
trouve en 1625 l'Instruction
de Sortie de l'Arithmétique.

A l'entrée dans le Sous-Programme de DIVISION, tant en
Simple qu'en Double Précision, le premier opérande (A,B)
est déjà chargé dans le PACx et le second opérande (A',B')
a été chargé dans le PDCx lors du passage par le Sous-Pro-
gramme d'Entrée.

5.1 DIVISION En Simple Précision.

Nous avons alors : (1619) = (1405)

5.1.1 Calcul du dénominateur du quotient

1405 65 1621 1576 (8002) = 00 1562 0000
1576 10 1479 1792 (8003) = 65 1621 1626

1792
A.S.D.P.
.. 1740

(PA) = ..A'..
(1740) = (1479)

1479 65 1621 1626 (8002) = 00 1562 0000
1626 10 1529 1790 (8003) = 21 1602 1606

1790
A.S.D.P.
.. 1740

Multiplication par A'
(PA) = A',2
(1740) = (1529)

1529 21 1602 1606 Transfert provisoire de A',2
en 1602.

1606 65 1517 1573 (8002) = 00 1567 0000
1573 10 1527 1792 (8003) = 65 1517 1623

1792
A.S.D.P.
.. 1740

(PA) = ..B'..
(1740) = (1527)

1527 65 1517 1623 (8002) = 00 1567 0000
1623 10 1627 1790 (8003) = 65 1578 1583

1790
A.S.D.P.
.. 1740

Multiplication par B'
(PA) = B',2
(1740) = (1627)

A.S.D.P.
.. 1740

(PA) = $A \times A' + B \times B'$
(1740) = (1469)

(8002) = 00 1604 0000
(8003) = 21 1550 1453

Division par $A'^2 + B'^2$
(PA) = $\frac{A \times A' + B \times B'}{A'^2 + B'^2}$
(1740) = (1432)
(1550) = partie réelle du quotient.

1789
A.S.D.P.
.. 1740

1432 21 1550 1453

5.1.4 Calcul de la partie imaginaire du quotient.

1453 65 1506 1611
1611 10 1564 1792

1792
A.S.D.P.
.. 1740

(PA) = $A' \times B - A \times B'$
(1740) = (1564)

1564 65 1574 1629
1629 10 1482 1789

1789
A.S.D.P.
.. 1740

Division par $A'^2 + B'^2$
(PA) = $\frac{A' \times B - A \times B'}{A'^2 + B'^2}$
(1740) = (1482)
(1555) = partie imaginaire du quotient.

En 1531 on retrouve l'instruction de Sortie de l'Arithmétique.

Partie réelle et imaginaire du quotient sont respectivement chargées en 1550 et 1555.

Remarquons que :
(1565) = (1570) = 0000000000

(8003) = 45 1440 1697

Addition de A'^2
(PA) = $A'^2 + B'^2$
(1740) = (1536)

On teste le dénominateur qui servira de diviseur dans la dernière partie du Sous-Programme.
S'il est nul, la suite est prise en 1697, à l'intérieur de l'ASDP et qui contient l'arrêt:

01 0222 8000

Transfert provisoire du dénominateur en 1604.

1440 21 1604 1507

5.1.2 Utilisation du SP de Multiplication.

Pour calculer la partie réelle et la partie imaginaire du quotient des 2 opérands, il faut effectuer le produit: $(A, B) \times (A', -B')$.

Pour cela, après avoir convenablement modifié le second opérande et adapté la sortie du SP de Multiplication, on prend la suite des instructions en 1406.

1507 61 1567 1524
1524 21 1567 1470
1470 69 1625 1628
1628 24 1531 1434
1434 69 1537 1490
1490 24 1625 1406

(1567) = $-B'$

(1531) = Instruction de Sortie.

(8) (1625) = (1537) = 65 1461 1416

1406
Sous-Programme de Multiplication
.. 1625

(1550) = $A \times A' + B \times B'$
(1555) = $A' \times B - A \times B'$
et (1625) = (1537)

5.1.3 Calcul de la partie réelle du quotient.

1537 65 1461 1416
1416 10 1469 1792

Nous avons alors : (1619) = (1423)

5.2 DIVISION en Double Précision

1423 65 1621 1640
1640 10 1493 1764

1764
A.S.D.P.
.. 1740

(PA) = ..A'..
(1740) = (1493)

1493 65 1621 1641
1641 10 1444 1761

1761
A.S.D.P.
.. 1740

Multiplication par A'
(PA) = A'^2
(1740) = (1444)

1444 21 1602 1559
1559 20 1617 1492
1492 65 1517 1542
1542 10 1495 1764

Transfert provisoire de A'^2
en 1602 et 1617.

1764
A.S.D.P.
.. 1740

(PA) = ..B'..
(1740) = (1495)

1495 65 1517 1592
1592 10 1545 1761

1761
A.S.D.P.
.. 1740

Multiplication par B'
(PA) = B'^2
(1740) = (1545)

1545 65 1578 1642
1642 10 1595 1757

1757
A.S.D.P.
.. 1740

Addition de A'^2
(PA) = A'^2 + B'^2 = (8003-2)
(1740) = (1595)

1595 45 1448 1697

On teste le dénominateur qui servira de diviseur dans la dernière partie du Sous-Programme.

prise à l'intérieur de l'ASDP dans la mémoire 1697 qui contient l'arrêt ;
01 0222 8000.

1448 21 1604 1609
1609 20 1619 1543

Transfert provisoire du dénominateur dans les mémoires 1604 et 1619

5.22 Utilisation du SP de Multiplication

Pour calculer la partie réelle et la partie imaginaire du quotient des 2 opérandes, il est nécessaire d'effectuer le produit :

$$(A, B) \times (A', -B')$$

Pour cela, après avoir convenablement modifié le second opérande et adapté la sortie du SP de Multiplication, on prend la suite des instructions en 1424.

1543 61 1567 1593
1593 16 1582 1643
1643 21 1567 1494
1494 20 1582 1544
1544 69 1625 1594
1594 24 1531 1644
1644 69 1447 1450
1450 24 1625 1424 (9)

(8003-2) = -B'
(1567) = -(B'...b'_1...)
(1582) = -(...b'_2....)
(1531)=(1625)= Inst.Sortie.
(1625)=(1447) = 65 1461 1645.

(9)

1424
Sous-Programme
de Multiplication
.. 1625

1550 et 1565 contiennent les 2 Mots de la quantité AxA' + BxB'
1555 et 1570 contiennent les 2 Mots de la quantité A'xB - AxB'
(1625) = (1447)

5.23 Calcul de la partie réelle du quotient.

1447 65 1461 1645
1645 10 1498 1764

(8002) = 00 1550 0000
(8003) = 65 1574 1446

1764
A.S.D.P.
.. 1740

(PA) = AxA' + BxB'
(1740) = (1498)

1498 65 1574 1446
1446 10 1499 1760

1760
A.S.D.P.
.. 1740

Division par $A'^2 + B'^2$
(PA) = $\frac{A \times A' + B \times B'}{A'^2 + B'^2}$
(1740) = (1499)

1499 21 1550 1503
1503 20 1565 1496

(1550) = Mot 1 de la partie réelle du quotient.
(1565) = Mot 2 de la partie réelle du quotient.

5.24 Calcul de la partie imaginaire du quotient.

1496 65 1506 1612
1612 10 1546 1764

(8002) = 00 1555 0000
(8003) = 65 1574 1596

1764
A.S.D.P.
.. 1740

(PA) = $A' \times B - A \times B'$
(1740) = (1546)

1546 65 1574 1596
1596 10 1549 1760

(8002) = 00 1604 0000
(8003) = 20 1570 1482

1760
A.S.D.P.
.. 1740

Division par $A'^2 + B'^2$
(PA) = $\frac{A' \times B - A \times B'}{A'^2 + B'^2}$
(1740) = (1549)

1549 20 1570 1482
1482 21 1555 1531

(1570) = Mot 2 de la partie imaginaire du quotient.
(1555) = Mot 1 de la partie imaginaire du quotient.

1531 Inst.Sortie

En 1531, on retrouve l'Instruction de Sortie de l'Arithmétique.

Partie réelle et partie imaginaire du quotient des 2 opérandes (A,B) et (A',B') sont chargées dans le PACx sous la forme :

(1550) = Mot 1 de la partie réelle
(1565) = Mot 2 de la partie réelle
(1555) = Mot 1 de la partie imaginaire.
(1570) = Mot 2 de la partie imaginaire.

ANNEXE 7

COMPILEUR MULTIPLE PRECISION

Pour l'interprétation et la traduction de tous les macro-codes définis au chapitre 3,
 Pour l'entrée, et d'une manière plus générale, pour l'utilisation, voir au Guide Opérateur,
 Section 3.3, page 121.

Liste des notations et abréviations particulières au Compileur :

- (A) : Adresse de l'instruction considérée - perforée dans les colonnes 43 à 47
- AB99 X : Constante symbolique interne
- C O P : Code OPération : colonnes 48 à 50
- (F) : adresse Facteur : colonnes 51 à 55
- (I) : adresse de l'Instruction suivante : colonnes 57 à 61
- M x x : Macro-code opération Multiple Précision
- REMARQUES : colonnes réservées aux remarques : 63 à 72
- X et X' : indicatifs de bibliothèque en colonne 51

- Code K : code fonctionnel double précision de la forme K F x x
- Code non K : code autre que fonctionnel, c'est-à-dire arithmétique
- Code PASO : code opération standard appartenant à la liste des codes directement interprétables par PASO III
- Code non PASO: macro-code non interprétable par PASO III
- Code V : macro-code arithmétique de la forme Vxx, opération en simple précision sur nombres réels
- Code non V : macro-code arithmétique autre que V x x
- Code ZZ ou MZZ: ce sont CZZ, DZZ, GZZ et TZZ
- Code non ZZ : autres macro-codes, de la forme M x x

- C.C.S. : sous-programme de Consultation de la Carte Suivante : analyse au 6.1 page 258.
- C.S.I. : sous-programme de construction de Constante Symbolique Interne : analyse au 6.2 page 259.

P.F.O. : sous-programme de numérotation et PerFOration : analyse au 6.3 page 260 .

TRANSCODAGE: pour représenter le transcodage d'une certaine quantité , nous écrirons cette quantité entre parenthèses , avec un indice t .

$C \dots)_t$: transcodage de la lettre C que , si nécessaire , nous expliciterons sous la forme

$$c_2 c_1 \quad 0000 \quad 0000$$

Mémoires à contenu variable.

1012	00 0000 00N°	Numéro de la carte perforée.
1031	Orientation	8888888888 : consultation de la carte suivante. 9999999999 : poursuite de la traduction.
1048	84 tttt 1600 1650 1324	Instruction de consultation de la Table des Codes sélectionnés. tttt = 1000 Table C = 1050 Table D = 1100 Table G = 1150 Table K = 1200 Table T = 1250 Table V
1099	{ 00 9197 9590 9999	Adresse de l'entrée dans la bibliothèque que pour un code arithmétique
1112	A B 9 9 X 00 909k 9k9k	Okkk est l'adresse de l'entrée dans la bibliothèque pour un code fonctionnel.
1139	X 0000 0000	Reçoit la constante symbolique interne AB99X après sa construction par le sous-programme C.S.I.
1187-1188-1189 1190-1191-1192		X est l'indicatif de bibliothèque perforé en colonne 51 de la carte BBL. Mémoires d'incompatibilité pour les codes fonctionnels. Chargées de 999999999+ s'il y a incompatibilité et de 000000000+ dans le cas contraire.
1194	69 pppp 1302	pppp est la première adresse de la zone du tambour correspondant à la fonction envisagée.
1242	(INSTR .)	L'adresse de la mémoire contenant
1324	{ 69 1721 1474 15 1166 1671	pppp est la première adresse de la zone du tambour correspondant à la fonction envisagée.
1366	000.. ou 999	Transfert provisoire de l'adresse I.
1368	000.. ou 999	Dans le cas d'une carte BBL de première espèce, ou bien
1370	000.. ou 999	Dans le cas d'une carte BBL de seconde espèce.
1372	000.. ou 999	Mémoire d'incompatibilité pour codes en C.
1374	000.. ou 999	Mémoire d'incompatibilité pour codes en D.
1424	10 1363+L 1389 ou bien 10 H+36 1389	Mémoire d'incompatibilité pour codes en G.
1431	{ 60 1500 1039 ou bien 60 1334 1039	Mémoire d'incompatibilité pour codes en K.
1450	00 0000 0mnp+	Contient le nombre des constantes symboliques internes construites par le sous-programme CSI.
1504	Contient l'instruction de sortie du sous-programme CSI.	
1558	69 rrrr 1452	rrrr est la dernière adresse de la zone du tambour correspondant à la fonction envisagée.
1560	Orientation	8888888888 : le macro-code de la carte lue est de la forme MZZ.
1592	Contient l'instruction de sortie du sous-programme PFO.	9999999999 : le macro-code de la carte lue n'est pas de la forme MZZ (code non-ZZ)
1610	Orientation	8888888888 : le code de la carte lue est un code standard PASO.
1642	Transfert provisoire	9999999999 : le code de la carte lue n'est pas un code standard PASO (code non-PASO).
1651	71 1.27 1592	Ordre de perforation de l'une des 10

1372	000.. ou 999	Mémoire d'incompatibilité pour codes en T.
1374	000.. ou 999	Mémoire d'incompatibilité pour codes en V.
1424	{ 10 1363+L 1389 ou bien 10 H+36 1389	Appel de la mémoire d'incompatibilité correspondant à la première lettre du code. L est le numéro d'ordre de la première lettre du code dans la table des premières lettres. Appel de la mémoire d'incompatibilité correspondant à la fonction considérée. H est l'adresse de l'argument de la Table K sur lequel s'est arrêtée la consultation.
1430	Orientation	8888888888 : le macro-code de la carte lue est un code K.
1431	{ 60 1500 1039 ou bien 60 1334 1039	9999999999 : le macro-code de la carte lue est un code non-K (C,D,G,T ou V) Entrée dans le sous-programme PFO pour perforation de la première carte de réservation correspondant à la première carte BBL de seconde espèce. Entrée dans le sous-programme PFO pour la perforation des autres cartes de réservation correspondant aux cartes BBL de seconde espèce.
1450	00 0000 0mnp+	Contient le nombre des constantes symboliques internes construites par le sous-programme CSI.
1504	Contient l'instruction de sortie du sous-programme CSI.	
1558	69 rrrr 1452	rrrr est la dernière adresse de la zone du tambour correspondant à la fonction envisagée.
1560	Orientation	8888888888 : le macro-code de la carte lue est de la forme MZZ.
1592	Contient l'instruction de sortie du sous-programme PFO.	9999999999 : le macro-code de la carte lue n'est pas de la forme MZZ (code non-ZZ)
1610	Orientation	8888888888 : le code de la carte lue est un code standard PASO.
1642	Transfert provisoire	9999999999 : le code de la carte lue n'est pas un code standard PASO (code non-PASO).
1651	71 1.27 1592	Ordre de perforation de l'une des 10

zones de sortie.
 1.27 est l'une des 10 adresses suivantes : 1527,1577,1627,1677,1727,1777,1827,1877,1927 ou 1977.

1692 Transfert provisoire.

1709 Orientation
 888888888 : le macro-code de la carte lue est un code en V.
 999999999 : le macro-code de la carte lue n'est pas un code en V (code non-V).

1719 20 1.35 1615
 Transfert du numéro de la carte sur le Mot 9 de la zone à perforer.
 1.35 est l'une des 10 adresses suivantes : 1535,1585,1635,1685,1735,1785,1835,1885,1935 ou 1985.

1742 Transfert provisoire.

1755 Contient l'instruction de sortie du sous-programme CCS.
 1756 Voir 1099

Autres mémoires à contenu variable:

Ce sont les mémoires de la zone de lecture et des 10 zones de sortie, exception faite des Mots à contenu fixe. (Voir le schéma des zones de sortie au §).

Mémoires à contenu fixe.

Ce sont, d'une part les mémoires chargées des différentes constantes utilisées par les différents sous-programmes, et d'autre part, les tables auxquelles il est fait appel en cours de compilation.

Mémoires chargées de constantes.

1016 00 9197 9090+ = (1700)_t Première mémoire de la zone correspondant à la bibliothèque A.

1018 69 1721 1474

1033 50 0013 0001+

1113 15 1166 1671

1114 21 0008 1194 +

1118 00 9195 9090 +

1132 60 1500 1039

1135 00 9090 9590+

1148 71 1577 1592

1166 00 0036 0000 +

1168 89 8900 0000 +

1169 00 9196 9590

1198 71 1727 1592

1265 65 0020 1700

1303 99 9999 9999 +

1306 90 9390 9090 +

1308 00 0012 0000 +

1312 62 6273 0000

1320 00 9195 9490

1326 00 0062 6273

1332 89 0000 0000

1394 10 0000 0000 +

1444 61 0000 0000 +

1470 00 9098 9495

1492 71 1527 1592

1503 60 1334 1039

1508 00 0008 0000 +

1521 00 9194 9090

1525 00 0000 0002 +

1526 71 1877 1592

1529 00 9199 9999

=(1500)_t Première adresse de la zone correspondant à la bibliothèque E.
 =(0050)_t Première adresse de la zone correspondant aux fonctions double précision.
 =(ZZ)_t
 =(1650)_t Première adresse de la zone correspondant aux bibliothèques B et D.
 =(BBL)_t
 =(1540)_t Première adresse de la zone correspondant à la bibliothèque C.
 =(BEL)_t
 =(Z)_t
 =(0845)_t Dernière adresse de la zone correspondant aux fonctions double précision.
 =(1400)_t Première adresse de la zone correspondant à la bibliothèque F.
 =(1999)_t Dernière adresse de la zone correspondant aux bibliothèques A, B, C, D, E et F.

1530	77 7961 0000	=(PRA) _t Mot fixe sur zone de sortie.
1542	71 1977 1592	
1550	00 0020 0000 +	
1551	71 1777 1592	
1575	00 0100 0038 +	
1580	65 6469 0000	=(EDI) _t Mot fixe sur zone de sortie.
1604	00 0000 9099 +	
1608	10 0013 1389 +	
1625	91 9300 0000 +	
1630	00 9090 0000	=(00) _t Mot fixe sur zone de sortie.
1647	90 9399 9999 +	
1654	71 1627 1592	
1664	71 1927 1592	
1680	79 6164 0000	=(RAD) _t Mot fixe sur zone de sortie.
1691	00 0000 0400 +	
1694	00 0100 0000 +	
1701	00 9090 9090 +	
1704	71 1677 1592	
1711	88 8888 8888 +	
1716	61 7583 6582	=(ANTES) _t
1720	61 6167 0000	=(AAD) _t Mot fixe sur zone de sortie.
1724	71 1827 1592	
1767	00 0000 0001 +	
1770	90 0000 0000 +	
1776	61 7583 6500	=(ANTE) _t
1780	79 6164 0000	=(RAD) _t Mot fixe sur zone de sortie.
1784	00 9090 0000	=(00) _t Mot fixe sur zone de sortie.
1800	77 7961 0000	=(PRA) _t Mot fixe sur zone de sortie.
1830	77 6483 0000	=(PDT) _t Mot fixe sur zone de sortie.
1931	63 7675 8283	=(CONST) _t Mot fixe sur zone de sortie.

TABLES

TABLE C

Table des codes opérations commençant par la lettre C

1000	63 6160 9999	Borne inférieure de la table
1001	63 6161 0000	=(CAA) _t
1002	63 6182 0000	=(CAS) _t
1003	63 6485 0000	=(CDV) _t
1004	63 7484 0000	=(CMU) _t
1005	63 8361 0000	=(CTA) _t : code standard PASO
1006	63 8379 0000	=(CTR) _t
1007	63 8961 0000	=(CZA) _t
1008	63 8982 0000	=(CZS) _t
1009	63 8989 0000	=(CZZ) _t
1010	99 9999 9999	Borne supérieure de la table.

Table des entrées pour codes C

1020	00 0000 0000	
1021	00 9197 9590	=(1750) _t
1022	00 9197 9591	=(1751) _t
1023	00 9197 9592	=(1752) _t
1024	00 9197 9593	=(1753) _t
1025	00 0000 0000	Correspond à un code standard PASO.
1026	00 9197 9594	=(1754) _t
1027	00 9197 9595	=(1755) _t
1028	00 9197 9596	=(1756) _t
1029	00 9197 9996	=(1796) _t
1030	00 0000 0000	

TABLE D

Table des codes opérations commençant par la lettre D

1050	64 6160 9999	Borne inférieure de la table
1051	64 6161 0000	=(DAA) _t
1052	64 6164 0000	=(DAD) _t : code standard PASO
1053	64 6167 0000	=(DAG) _t : code standard PASO
1054	64 6182 0000	=(DAS) _t
1055	64 6185 0000	=(DAV) _t
1056	64 6461 0000	=(DDA) _t : code standard PASO
1057	64 6485 0000	=(DDV) _t
1058	64 6763 0000	=(DGC) _t : code standard PASO
1059	64 6985 0000	=(DIV) _t : code standard PASO.

1106	67 8379 0000	=(GTR) t
1107	67 8961 0000	=(GZA) t
1108	67 8982 0000	=(GZS) t
1109	67 8989 0000	=(GZZ) t
1110	99 9999 9999	Borne supérieure de la table.

Table des entrées pour codes G.

1120	00 0000 0000	
1121	00 9197 9698	=(1768) t
1122	00 9197 9699	=(1769) t
1123	00 9197 9790	=(1770) t
1124	00 9197 9791	=(1771) t
1125	00 0000 0000	Correspond à un code standard PASO
1126	00 9197 9792	=(1772) t
1127	00 9197 9793	=(1773) t
1128	00 9197 9794	=(1774) t
1129	00 9197 9998	=(1798) t
1130	00 0000 0000	

TABLE K

Table des codes fonctionnels commençant par la lettre K.

1150	72 6660 9999	Borne inférieure de la table.
1151	72 6661 0000	=(KFA) t: Arc tangente.
1152	72 6663 0000	=(KFC) t: Cosinus
1153	72 6665 0000	=(KFE) t: Exponentielle
1154	72 6673 0000	=(KFL) t: Logarithme népérien.
1155	72 6679 0000	=(KFR) t: Racine carrée.
1156	72 6682 0000	=(KFS) t: Sinus.
1157	99 9999 9999	Borne supérieure de la table.

Table des adresses de la première mémoire de la zone correspondant à chaque fonction.

1158	00 9999 9999	
1159	00 9096 9495	=(0645) t
1160	00 9094 9890	=(0480) t
1161	00 9093 9190	=(0310) t
1162	00 9091 9590	=(0150) t
1163	00 9090 9590	=(0050) t
1164	00 9094 9890	=(0480) t
1165	00 9999 9999	

Tables des entrées pour codes K

1060	64 7484 0000	=(DMU) t
1061	64 7979 0000	=(DER) t: code standard PASO
1062	64 8285 0000	=(DSV) t
1063	64 8379 0000	=(DTR) t
1064	64 8961 0000	=(DZA) t
1065	64 8982 0000	=(DZS) t
1066	64 8985 0000	=(DZV) t
1067	64 8986 0000	=(DZW) t
1068	64 8989 0000	=(DZZ) t
1069	99 9999 9999	Borne supérieure de la table D.

Table des entrées pour codes D

1070	00 0000 0000	
1071	00 9197 9597	=(1757) t
1072	00 0000 0000	Correspond à un code standard PASO
1073	00 0000 0000	Correspond à un code standard PASO
1074	00 9197 9598	=(1758) t
1075	00 9197 9599	=(1759) t
1076	00 0000 0000	Correspond à un code standard PASO
1077	00 9197 9690	=(1760) t
1078	00 0000 0000	Correspond à un code standard PASO
1079	00 0000 0000	Correspond à un code standard PASO
1080	00 9197 9691	=(1761) t
1081	00 0000 0000	Correspond à un code standard PASO
1082	00 9197 9692	=(1762) t
1083	00 9197 9693	=(1763) t
1084	00 9197 9694	=(1764) t
1085	00 9197 9695	=(1765) t
1086	00 9197 9696	=(1766) t
1087	00 9197 9697	=(1767) t
1088	00 9197 9997	=(1797) t
1089	00 0000 0000	

TABLE G

Table des codes opérations commençant par la lettre G.

1100	67 6160 9999	Borne inférieure de la table.
1101	67 6161 0000	=(GAA) t
1102	67 6182 0000	=(GAS) t
1103	67 6485 0000	=(GDV) t
1104	67 7484 0000	=(GDU) t
1105	67 7575 0000	=(GEM) t: code standard PASO

1220 00 0000 0000
 1221 00 9197 9795 =(1775) t
 1222 00 9197 9796 =(1776) t
 1223 00 9197 9797 =(1777) t
 1224 00 0000 0000 Correspond à un code standard PASO
 1225 00 9197 9798 =(1778) t
 1226 00 9197 9799 =(1779) t
 1227 00 0000 0000 Correspond à un code standard PASO.
 1228 00 0000 0000 Correspond à un code standard PASO.
 1229 00 9197 9890 =(1780) t
 1230 00 9197 9891 =(1781) t
 1231 00 9197 9892 =(1782) t
 1232 00 9197 9893 =(1783) t
 1233 00 9197 9894 =(1784) t
 1234 00 9197 9895 =(1785) t
 1235 00 9197 9999 =(1799) t
 1236 00 0000 0000

TABLE V

Table des codes opérations commençant par la lettre V.

1250 85 6160 9999 Borne inférieure de la table.
 1251 85 6161 0000 =(VAA) t
 1252 85 6182 0000 =(VAS) t
 1253 85 6185 0000 =(VAV) t
 1254 85 6485 0000 =(VDV) t
 1255 85 7484 0000 =(VMU) t
 1256 85 8285 0000 =(VSV) t
 1257 85 8961 0000 =(VZA) t
 1258 85 8982 0000 =(VZS) t
 1259 85 8985 0000 =(VZV) t
 1260 85 8986 0000 =(VZW) t
 1261 99 9999 9999 Borne supérieure de la table.

Table des entrées pour codes V

1270 00 0000 0000
 1271 00 9197 9896 =(1786) t
 1272 00 9197 9897 =(1787) t
 1273 00 9197 9898 =(1788) t
 1274 00 9197 9899 =(1789) t
 1275 00 9197 9990 =(1790) t
 1276 00 9197 9991 =(1791) t

1170 00 0000 0000
 1171 00 9097 9090 =(0700) t
 1172 00 9095 9091 =(0501) t
 1173 00 9093 9590 =(0350) t
 1174 00 9091 9590 =(0150) t
 1175 00 9090 9590 =(0050) t
 1176 00 9095 9090 =(0500) t
 1177 00 0000 0000

Table des adresses de la dernière mémoire de la zone correspondant à chaque fonction.

1178 00 9999 9999
 1179 00 9098 9494 =(0844) t
 1180 00 9096 9494 =(0644) t
 1181 00 9094 9799 =(0479) t
 1182 00 9093 9099 =(0309) t
 1183 00 9091 9499 =(0149) t
 1184 00 9096 9494 =(0644) t
 1185 00 9999 9999

TABLE T

Table des codes opérations commençant par la lettre T.

1200 83 6160 0000 Borne inférieure de la table.
 1201 83 6161 0000 =(TAA) t
 1202 83 6182 0000 =(TAS) t
 1203 83 6185 0000 =(TAV) t
 1204 83 6489 0000 =(TDI) t: Code standard PASO.
 1205 83 6485 0000 =(TDV) t
 1206 83 7484 0000 =(TMU) t
 1207 83 7964 0000 =(TRD) t: Code standard PASO
 1208 83 7967 0000 =(TRG) t: Code standard PASO
 1209 83 8285 0000 =(TSV) t
 1210 83 8379 0000 =(TTR) t
 1211 83 8961 0000 =(TZA) t
 1212 83 8982 0000 =(Tzs) t
 1213 83 8985 0000 =(TZV) t
 1214 83 8986 0000 =(TZW) t
 1215 83 8989 0000 =(TZZ) t
 1216 99 9999 9999 Borne supérieure de la table.

Table des entrées pour codes T.

1400	60 9999 9999	Borne inférieure de la table.
1401	61 0000 0000	=(A) _t
1402	62 0000 0000	=(B) _t
1403	63 0000 0000	=(C) _t
1404	64 0000 0000	=(D) _t
1405	65 0000 0000	=(E) _t
1406	66 0000 0000	=(F) _t
1407	67 0000 0000	=(G) _t
1408	68 0000 0000	=(H) _t
1409	69 0000 0000	=(I) _t
1410	71 0000 0000	=(J) _t
1411	99 9999 9999	Borne supérieure de la table

1277	00 9197 9992	=(1792) _t
1278	00 9197 9993	=(1793) _t
1279	00 9197 9994	=(1794) _t
1280	00 9197 9995	=(1795) _t
1281	00 0000 0000	

Table des premières lettres des codes opérations

1350	62 9999 9999	
1351	63 0000 0000	=(C) _t
1352	63 9999 9999	
1353	64 0000 0000	=(D) _t
1354	66 9999 9999	
1355	67 0000 0000	=(G) _t
1356	71 9999 9999	
1357	72 0000 0000	=(K) _t
1358	82 9999 9999	
1359	83 0000 0000	=(T) _t
1360	84 9999 9999	
1361	85 0000 0000	=(V) _t
1362	99 9999 9999	

Instructions pour consultation des Tables C, D, G, K, T ou V

1376	00 0000 0000	Le code opération n'est pas un macro-code.
1377	84 1000 1600	Le code est un code C. Consultation de la TABLE C.
1378	00 0000 0000	Le code n'est pas un macro-code.
1379	84 1050 1600	Le code est un code en D. Consultation de la TABLE D.
1380	00 0000 0000	Le code n'est pas un macro-code.
1381	84 1100 1600	Le code est un code en G. Consultation de la TABLE G.
1382	00 0000 0000	Le code n'est pas un macro-code.
1383	84 1150 1324	Le code est un code en K. Consultation de la TABLE K.
1384	00 0000 0000	Le code n'est pas un macro-code.
1385	84 1200 1600	Le code est un code en T. Consultation de la TABLE T.
1386	00 0000 0000	Le code n'est pas un macro-code.
1387	84 1250 1650	Le code est un code en V. Consultation de la TABLE V.
1388	00 0000 0000	Le code n'est pas un macro-code.

Table des indicatifs de bibliothèques

§ 1 REINITIALISATION

1.1 Remise à zéro de la comptation des cartes.

1300 65 1303 1307 (8002) = 9999999999 +
 (8003) = 0000000000 +
 1307 21 1012 1015 (1012) = 0000000000 +

On remet à zéro les Mots 9 contenant le numéro de la carte perfo-
 forée sur les 10 zones de perforation.

1015 24 1985 1038
 1038 24 1935 1138
 1138 24 1885 1238
 1238 24 1835 1288
 1288 24 1785 1338
 1338 24 1735 1438
 1438 24 1685 1488
 1488 24 1635 1538
 1538 24 1585 1588
 1588 24 1535 1638

1.2 Remise à zéro des mémoires d'incompatibilité.

1638 20 1364 1017 (8002)=(8001) = 9999999999 +

1017 24 1374 1327
 1327 24 1366 1019
 1019 24 1368 1321
 1321 24 1372 1325
 1325 24 1370 1323
 1323 24 1187 1040
 1040 24 1188 1041
 1041 24 1189 1042
 1042 24 1090 1043
 1043 24 1191 1044
 1044 24 1192 1045

1.3 Initialisation du sous-programme C.S.I.

1045 21 1450 1453 (1450) = 0000000000 +
 Remise à zéro de la mémoire de comptage des constantes symboliques internes.

1453 69 1306 1309
 1309 24 1112 1115

(1112) = 9093909090 = (03000)^t
 Constante symbolique interne initiale.

1.4 Autres réinitialisations.

1115 69 1018 1471
 1471 24 1324 1427 (1324) = 69 1721 1474
 1427 20 1031 1034 (1031) = 9999999999 +

§ 2 TRADUCTION DES PSEUDO-INSTRUCTIONS BBL.

2.1 Carte BBL de première espèce.

2.1.1 Elimination des cartes non-BBL de première espèce

1034 70 1963 1961 En 1963, la carte étant chargement, on trouve l'arrêt

01 0333 8000

(12) 1961 65 1965 1459
 1459 35 0006 1473
 1473 11 1326 1131
 1131 44 1035 1036

(8003) = (COP)^t

(1326) = (BBL)^t

(8003) = COP - BBL ≠ 0 ?

En 1035, puisque le code opération n'est pas BBL, la première est erronée et on trouve l'arrêt : 01 0301 1300

(8002) = colonnes 57 à 61

On teste ce contenu. S'il est nul, les colonnes 57 à 61 n'ayant donc pas été perforées, la carte BBL est de première espèce; dans le cas contraire, elle est de seconde espèce.

1036 65 1953 1457
 1457 45 1310 1011
 (1) (2)

(2)

1011 60 1326 1331
 1331 15 1952 1507
 1507 35 0004 1117
 1117 21 1531 1134
 1134 21 1881 1496

(8003) = (BBLX)^t

L'indicatif BBLX est transféré sur Mot 5, en Remarques des zones de sortie correspondant à une carte BBL.

On charge 0000000000 + en Mot 6 de ces mêmes zones.

1496 20 1532 1498
 1498 24 1882 1284

2.12 Sélection de la bibliothèque

1284 65 8002 1093 (8003-2) = 000 ... 00
 1093 69 1952 1305 (8001) = X_t 0000 0000
 1305 84 1400 1455 Consultation de la table des
 indicatifs de bibliothèque.
 (8002) = 00 (1400 + a) 0000
 avec 00 < a < 11.
 (8002) = 00 (1412 + a) 8880

Suivant la valeur de a, donc la
 nature de l'indicatif, la suite
 sera prise entre 1412 et 1423.

2.13 Réserve des zones de bibliothèques et libération des mémoires d'incompatibilité.

1- Bibliothèques inconnues: a=0 ou a=11

1412 01 0302 1300 X est inférieur à A
 1423 01 0302 1300 X est supérieur à J, l'indicatif
 ne correspond à aucune bibliothèque
 que connue du Compilateur.

2- Bibliothèque A: a=1, codes D

(6) 1413 60 1016 1621 (8003) = 0091979090 = (1790)^t
 Première adresse de la zone de
 la bibliothèque A.
 1621 20 1366 1571 (3) (1366) = 0000000000 +
 Libération de la mémoire d'in-
 compatibilité pour codes D.

3- Bibliothèque B: a=2, codes D et V.

(7) 1414 60 1169 1623 (8003) = 0091969590 = (1650)^t
 Première adresse de la zone de
 la bibliothèque B.
 1623 20 1366 1217 (1366) = 0000000000 +
 1217 20 1374 1571 (3) (1374) = 0000000000 +
 Libération des mémoires d'incom-
 patibilité pour codes D et V.

4- Bibliothèque C: a=3, codes I

1415 60 1118 1573 (8003) = 0091959590 = (1550)^t
 Première adresse de la zone de
 la bibliothèque C.
 1573 20 1372 1571 (3) (1372) = 0000000000 +
 Libération de la mémoire d'incom-
 patibilité pour codes T.

5- Bibliothèque D: a=4, codes C

1416 60 1169 1523 (8003) = 0091969590 = (1650)^t

Première adresse de la zone de
 la bibliothèque D.

1523 20 1364 1571 (3) (1364) = 0000000000 +
 Libération de la mémoire d'in-
 compatibilité pour codes C.

6- Bibliothèque E: a=5, codes D et G

(8) 1417 60 1320 1477 (4) (8003) = 0091959490 = (1540)^t
 Première adresse de la zone de
 la bibliothèque D.

7- Bibliothèque F: a=6, codes C, D, G et V

(9) 1418 60 1521 1425 (8003) = 0091949090 = (1400)^t
 1425 20 1364 1167 (1364) = 0000000000 +
 1167 20 1374 1477 (1374) = 0000000000 +
 Libération des mémoires d'incom-
 patibilité pour codes C et V.

(4) 1477 20 1366 1119 (1366) = 0000000000 +

1119 20 1368 1571 (1368) = 0000000000 +
 Libération des mémoires d'incom-
 patibilité pour codes D et G.

(3) 1571 21 1528 1431 (1528) = Première adresse de la
 zone correspondant à la biblio-
 thèque considérée, chargée en
 Mot 2 de la zone de perforation.
 1431 60 1334 1039 (8003) = (1334)

Numérotation et perforation de
 la carte de réservation !

PRA 1xxx 1999 BBL X

(1592) = (1334) (38)

La suite est prise en 1334 pour
 la lecture de la seconde carte
 de la liste à compiler.

8- Bibliothèque G: a=7, codes K et BBL A

1419 60 1413 1267 (5) (8003) = (1413)

9- Bibliothèque H: a=8, codes K et BBL R

1420 60 1414 1267 (5) (8003) = (1414)

10- Bibliothèque I: a=9, codes K et BBL E

1421 60 1417 1267 (5) (8003) = (1417)

11- Bibliothèque J: a=10, codes K et BBL F

1422 60 1418 1267 (5) (8003) = (1418)

(5) 1267 69 1470 1673

1673 24 1879 1032

(1879) = (0845)^t, deuxième a-
 dresse de la zone correspondant

(8002) = perforation des colonnes 57 à 61 = (KFXx)_t
 Transfert de KFXx en Mot 6 des zones de sortie correspondantes.
 (8002) = (X)_t, indicatif de bibliothèque.
 Transfert de X en Mot 5 des zones de sortie correspondantes.
 (1139) = (X)_t
 (8002) = X - A

Puisque X = A, la suite est prise en 1413 pour perforation de la carte de réservation :

PRA 1700 1999 A KFXx

(8002) = X - B

Puisque X = B, la suite est prise en 1414 pour la perforation de la carte de réservation :

PRA 1650 1999 B KFXx

(8002) = X - E

Puisque X = E, la suite est prise en 1417 pour perforation de la carte de réservation :

PRA 1540 1999 E KFXx

(8002) = X - F

Puisque X = F, la suite est prise en 1418 pour perforation de la carte de réservation :

PRA 1400 1999 F KFXx

aux fonctions double précision.

(1878) = (0050)_t, première adresse de la zone correspondant aux fonctions double précision.

2878 s8 1870 étant les Mots 2 et 3 de la zone de sortie correspondant.

(1370) = 0000000000 + libération de la mémoire d'incompatibilité pour codes K.

Numérotation et perforation de la carte de réservation :

PRA 0050 0845 BBL X

La suite est prise en : 1413 (6), 1414 (7), 1417 (8) ou 1418 (9), suivant la bibliothèque pour perforation de la carte de réservation de la zone correspondant à la bibliothèque A, B, E ou F. (Voir ci-dessus).

2.2 Carte BBL de seconde espèce.

2.2.1 Adaptations.

(1) 1310 69 1113 1116

1116 24 1324 1328

1328 69 1132 1285

1285 24 1431 1434

(1324) = 15 1166 1671
 Modification de la recherche des incompatibilités pour codes K.

(1431) = 60 1500 1039
 Modification de la sortie pour une carte BBL de première espèce. On aura :

(3) 1571 21 1528 1431
 1431 60 1500 1039

1039
 P.F.O.
 1592

(1592) = (1500)

2.2.2 Recherche de la bibliothèque arithmétique

1032 69 1135 1688

1688 24 1878 1481

1481 20 1370 1423

1423
 P.F.O.
 1592

1434 20 1532 1335

1335 24 1062 1435

1435 69 1952 1557

1557 20 1531 1484

1484 24 1881 1485

1485 20 1139 1092

1092 16 1401 1505

1505 45 1458 1413

(6)

(6) 1413

.... 1500 (10)

1458 15 8001 1317

1317 16 1402 1607

1607 45 1460 1414

(7)

(7) 1414

.... 1500 (10)

1460 15 8001 1219

1219 16 1405 1509

1509 45 1262 1417

(8)

(8) 1417

.... 1500 (10)

1262 15 8001 1771

1771 16 1406 1111

1111 45 1014 1418

(9)

(9) 1418

.... 1500 (10)

X est différent de A,B,Z ou F.
L'indicatif de bibliothèque est donc incompatible avec l'utilisation de fonctions double précision.

2.23 Recherche de la fonction et réservation

(10) 1500 69 1503 1456

1456 24 1431 1136

1136 65 1953 1657

1657 30 0004 1467

1467 35 0004 1329

1329 16 8002 1037

1037 84 1150 1555

1555 15 1508 1263

1263 69 1194 1047

1047 22 1194 1097

1097 15 1550 1605

1605 69 1558 1311

1311 22 1558 1461

1461 15 1114 8002

8002 21 1186 1194

+b

1194 69 1158 1302

+b

1302 24 1878 1558

1558 69 1178 1452

+b

(1431) = 60 1334 1039

Réinitialisation de la sortie pour une carte BBL de première espèce.

(8002) = (KFX) ^t

(8002) = 0000000000 +

(8001) = (KFX) ^t

Consultation de la TABLE X.

(8002) = 00 (1150+b) 0000

avec 0 < b < 7

(8002) = 00 (1158+b) 0000

(8001) = 69 xxx 1302

(1194) = 69 (1158+b) 1302

(8002) = 00 (1178+b) 0000

(8001) = 69 xxx 1452

(1558) = 69 (1178+b) 1452

(8003) = 0000000000 +

Libération de la mémoire d'incompatibilité correspondant à la fonction :

KFAT : b=1, mémoire 1187

KFCN : b=2, mémoire 1188

KFEX : b=3, mémoire 1189

KFLE : b=4, mémoire 1190

KFRC : b=5, mémoire 1191

KFSN : b=6, mémoire 1192

1878 reçoit la première adresse de la zone correspondant à la fonction :

b = 1, (1878) = (0645) ^t

b = 2, (1878) = (0480) ^t

b = 3, (1878) = (0310) ^t

b = 4, (1878) = (0150) ^t

b = 5, (1878) = (0050) ^t

b = 6, (1878) = (0480) ^t

1879 reçoit la dernière adresse

1452 24 1879 1282

de la zone correspondant à la fonction :

b = 1, (1879) = (0844) ^t

b = 2, (1879) = (0644) ^t

b = 3, (1879) = (0479) ^t

b = 4, (1879) = (0309) ^t

b = 5, (1879) = (0149) ^t

b = 6, (1879) = (0644) ^t

1282 60-1286 1723

(8003) = (1286)

1723

P.F.O.

.... 1592

Numérotation et perforation de la carte de réservation correspondant à la fonction considérée :

PRA 0... 0... X KFX

(1592) = (1286)

2.24 Lecture de la carte suivant la première carte BBL de première espèce.

1286 70 1963 1964

(1963) = 0103338000, arrêt pour carte chargement.

1964 65 1954 1559

(8002) = (COP) ^t

1559 16 1312 1517

(8002) = COP - BBL

1517 45 1962 1322

En 1962, le code n'étant pas BBL, la suite est prise pour la recherche de ce code.

(11)

1322 65 1953 1707

(8002) = contenu des colonnes 57 à 61.

1707 45 1510 1961

Si les colonnes sont blanches, la carte BBL est de première espèce. La suite est prise en 1961.

(12)

1510 65 1952 1757

(8002) = (X') ^t, indicatif de bibliothèque de la nouvelle carte BBL de seconde espèce.

1757 16 1139 1143

(8002) = X' - X

1143 45 1046 1147

Si X' ≠ X, c'est à dire si les indicateurs de bibliothèque des 2 cartes BBL de seconde espèce consécutives ne sont pas identiques, on trouve l'arrêt programmé :

(1406) = 01 0344 8000

1147 15 8001 1546

1546 69 1953 1548

Transfert de KFX sur Mot 6 de la zone de sortie.

1548 24 1882 1500

(10)

La suite est prise en 1500 pour la recherche de la nouvelle fonction considérée et la perforation de la réservation correspondante.

1579 reçoit la dernière adresse

§ 3 TRADUCTION D'UNE
MACRO-INSTRUCTION

3.1 Interprétation d'un code-opération Mix

3.1.1 Elimination des cartes BBL et Z

(38) 1374 70 1963 1962 (1963) = 01 0333 8000, cas d'une
(11) (11) carte chargement.
1609 65 1954 1609 (8002) = (COP) _t
1567 45 1520 1472 (8002) = COP - BBL
Si la carte est une carte BBL de
première espèce, on trouve l'ar-
rêt programmé :
(1472) = 01 0303 1334
(1472) = 01 0303 1334
1520 15 8001 1429 (8002) = COP - Z
1429 16 1332 1137
1137 45 1090 1091 Si COP = Z, la suite est prise
(32) en 1091 pour la traduction de
la carte Z.

3.1.2 Recherche de la première lettre du code
Première séparation des codes PASO.

1090 15 8001 1049
1049 35 0002 1655 (8003) = 00000000(0)_t, première
lettre du COP.
1655 60 8003 1313 (8002) = (0)_t 0000 0000
1313 30 0002 1269 (8002) = 0000000000 +
1269 16 8002 1479 (8001) = (C)_t 0000 0000
1479 84 1350 1705 Consultation de la table des pre-
mières lettres du code-opération.
(8002) = 00 (1350+L) 0000
Si L = 0, 2, 4, 5, 8, 10 ou 12, la
lettre n'appartient pas à la ta-
ble.
Si L = 1, première lettre = C
Si L = 3, première lettre = D
Si L = 5, première lettre = G
Si L = 7, première lettre = K
Si L = 9, première lettre = T
Si L = 11, première lettre = V
(8002) = 10 (1363+L) 1389
(1424) = 10 (1363+L) 1389, in-
truction d'appel de la mémoire
1705 15 1608 1463
1463 20 1424 1350

d'incompatibilité pour le codes
considérés.

1330 15 1033 8002 (1033) = 50 0013 0001
8002 60 1376 1390 Si L = 0, 2, 4, 6, 8, 10 ou 12, le
contenu de la mémoire correspon-
dante est nul, donc, dans ce cas,
(8003) = 0000000000 +

Dans le cas contraire, la mémoi-
re considérée contient l'instruc-
tion de consultation de la table
correspondant au code.

L = 1, (8003) = 84 1000 1600
L = 3, (8003) = 84 1050 1600
L = 5, (8003) = 84 1100 1600
L = 7, (8003) = 84 1150 1324
L = 9, (8003) = 84 1200 1600
L = 11, (8003) = 84 1250 1650
Si (8003) = 0, la première lettre
du code n'étant pas l'une des 6
lettres considérées, le code est
un code standard PASO; la suite
est prise en 1094.

Dans le cas contraire il faut
poursuivre la recherche à l'in-
térieure de la table sélection-
née.

3.1.3 Seconde recherche du code.

1243 21 1048 1301 (1048) = consultation de la table
1301 65 8002 1659 (8003-2) = 000 ... 000 +
1659 69 1954 1048 (8001) = (COP) _t
1048 84 1000 1650 (8002) = 00 H 0000
à 1324 H étant l'adresse de l'argument
de la table sur lequel s'est
arrêté la consultation.
1250 1600

1er Cas: codes V

1650 69 1303 1556
1556 24 1430 1283 (1430) = 9999999999+
1283 69 1721 1506 (15) (8001) = 8888888888+

2ième Cas: codes K avec BBL de première espèce.

1324 69 1721 1474 (14) (8001) = 8888888888+

3ème Cas: codes K avec BBL de seconde espèce.

1324 15 1166 1671 (8002) = 00 (H+36) 0000
1671 69 1424 1428
1428 22 1424-1478 (1424) = 10 (H+36) 1389
Appel de la mémoire d'incompati-
bilité du code fonctionnel.
1478 16 1166 1018 (8002) = 00 H 0000
1018 69 1721 1474 (14) (8001) = 8888888888+

4ème cas : Codes C, D, G et H

- 1600 69 1303 1474
(14) 1474 24 1430 1133
= 999999999+ pour les codes C, D, G et H et V
= 888888888+ pour les codes K
1133 69 1303 1506
(15) 1506 24 1709 1462
(1709) = 999999999+ pour les codes C, D, G, K et T
= 888888888+ pour le code V.
1462 15 1265 8002
(8002) = 65 (H+20) 1700

3.14 Seconde séparation des codes PASO incompatibilités.

- 8002 65 H+20 1700
(8002) = 000000000+ pour les codes standards PASO
= 00 9197 9999 ou 00 909k 9k9k : adresse de l'entrée dans la bibliothèque pour les macro-codes.
1700 45 1424 1094
(15)
1424 10 1363 1359
+L
ou bien
1424 10 H+36 1389
1389 44 1293 1144

Appel de la mémoire d'incompatibilité correspondant au code fonctionnel considéré.
On teste donc le contenu de 8003
Si (8003) = 999999999+, il y a incompatibilité entre le code Mxx présentement considéré et la bibliothèque envisagée (ou bien les fonctions envisagées)
On trouve donc l'arrêt :
(1293) = 01 0340 8000

3.15 Séparation des codes MZZ

- 1144 20 1099 1502
1502 60 1954 1759
1759 35 0002 1315
1315 11 1168 1773
(1099) = (17⁵⁰)_t ou (0kxx)_t
(8003) = (COP)_t
(8003) = (OP)_t
(8003) = OP - ZZ

- 1773 45 1426 1480
1426 69 1303 1606
1480 69 1721 1606
1606 24 1560 1513

3.16 Orientations

- 1513 69 1303 1656
(13) 1094 69 1721 1656
1656 24 1610 1563
1563 69 1031 1336
1336 90 1140 1141
(17)
(8001) = 999999999+ + code non-PASO
(8001) = 888888888+ + code PASO
(1610) = 88.8 ou 99.9 suivant que le code est PASO ou non.
(8001) = 88.8 ou 99.9
Si 8001 est chargé de 8, cela signifie que la carte présente-ment analysée ne l'est que pour consultation.
Si 8001 est chargé de 9, l'analyse se poursuit normalement par la traduction de la macro-instruction.

- 1140 Voir le sous-Programme de Consultation de la Carte Suivante (C.C.S.)

3.2 Traduction d'une carte PASO

3.21 Libération de la zone de lecture

- (17) 1141 60 1951 1706
1706 21 1977 1432
1432 60 1952 1658
1658 21 1978 1482
1482 15 1953 1708
1708 20 1979 1333
1333 69 1954 1758
1758 84 1980 1433
1433 69 1099 1552
1552 24 1756 1660
1660 69 1610 1613
1613 90 1617 1218
(18) (19)
(8003) = Adresse A
Transférée en Mot 1, (1977) = A
(8003) = Adresse facteur F
Transférée en Mot 2, (1978) = F
(8002) = Instruction suivante I
Transférée en Mot 3, (1979) = I
(1980) = COP
(1756) = 17⁵⁰ ou 0kxx
(8001) = 888.8 pour carte PASO
= 999.9 pour carte nom-PASO.

3.22 Cas où l'adresse F est blanche

(18) 1617 44 1522 1572 (20) (8003) = adresse facteur F

1572 45 1476 1483 (8002) = adresse I
Si F et I sont simultanément blancs, on trouve l'arrêt :

(1483) = 01 0310 1334 pour erreur dans l'enchaînement des instructions.

1476 69 1436 1239 (8001) = (1436)

1239
C.C.S.
.... 1755

(8003) = A_s et (1755) = (1436)

1436 69 1610 1663 (8001) = 88.8 ou 99.9

1663 90 1667 1268 88.8 si la carte consultée est PASO,
99.9 si la carte consultée n'est pas PASO

(21)

Code non-PASO pour la carte consultée

1268 44 1622 1672 Si A est non-blanc alors que F l'est, il y a erreur d'enchaînement des instructions, d'où l'arrêt :

(1622) = 01 0320 1493

1672 69 1430 1486 (8001) = 88.8 si le code de la carte consultée est un code K = 99.9 si le code de la carte consultée est non-K.

1486 90 1667 1241 (21)

1241 60 1244 1149 (8003) = (1244)

1149 C.S.I. 1504

1244 21 1951 1304 Puisque le code est non-K et que I et A sont blancs, il faut leur attribuer le même équivalent symbolique.

1304 21 1978 1667 (21) (8003) = (AB99X)_t
(1504) = (1244)
(1951) = (AB99X)_t
(1978) = (AB99X)_t

Code PASO pour la carte consultée.

(21) 1667 69 1303 1760

1760 24 1031 1337

(1031) = 9999999999+
Charge 1031 de 9 pour permettre de poursuivre la traduction de la carte qui vient d'être consultée.

(8003) = (1141)

1337 60 1441 1289 (17)

1289
P.F.O.
.... 1592

Numérotation et perforation de la carte PASO:

(A) COP AB99X (I)

(1592) = (1141) (17)

Du fait de la consultation de carte suivante, la suite des instructions est prise en 1141 où l'on poursuit la traduction de la carte consultée.

3.23 Cas où l'adresse F est non blanche.

(20) 1522 60 8002 1237 (8003) = I

1237 44 1291 1142 (22)

Cas où I est également non-blanc

1291 60 1334 1289 (8003) = (1334)

(38)

1289
P.F.O.
.... 1592

Numérotation et reproduction de la carte PASO:

(A) COP (F) (I)

(1592) = (1334) (38)

Puisqu'il n'y a pas eu consultation de la carte suivante, la suite est prise en 1334 pour lecture de la carte suivante.

Cas où I est blanc :

(22) 1142 69 1095 1239 (8001) = (1095)

1239
C.C.S.
.... 1755

(8003) = A_s

(1755) = (1095)

(8001) = 88.8 si la carte suivante est PASO

= 99.9 si la carte suivante est non-PASO.

. 253.

1713 90 1667 1318 (21)

1318 44 1622 1722 (8003) = A_s

Si A est blanc alors que I ne l'est pas, il y a erreur dans l'enchaînement, d'où l'arrêt :
(1622) = 01 0320 1493

(8001) = 88.8 si le code de la carte consultée est K
= 99.9 s'il est non-K

(8003) = (1145)

(8003) = AB99W
(1504) = (1145)
(1551) = AB99W

(1979) = AB99W, sur Mot 3 de la zone de perforation en remplacement de I qui était blanc sur la carte lue.

(8001) = 888888888 + pour la carte K.

(x) transféré en Mot 6 sur la zone de sortie.
(Kx) transféré en Mot 5.
(1579) = Okkk, en Mot 3 de la zone de sortie.
(1577) = Adresse A, en Mot 1 de la zone de sortie.

1722 69 1430 1287
1287 90 1667 1242 (21)
1242 60 1145 1149

1149
C.S.I.
.... 1504

1145 21 1951 1454
1454 21 1979 1667 (21)

Nous avons : (1610) = 999999999 +
(1430) = 888888888 +

Une carte K de la forme (A) KFx x (I)
est traduite par (A) EDI (I) Okkk

Okkk étant l'entrée dans le bibliothèque fonctionnelle correspondant au code Kfix.

3.31 Séparation et préparations

(19) 1218 69 1430 1437
1437 90 1341 1292 (23)
1341 21 1582 1487
1487 69 1980 1537
1537 24 1581 1587
1587 69 1756 1511
1511 24 1579 1637
1637 69 1977 1687
1687 24 1577 1737

(8001) = 888888888 + pour la carte K.
(x) transféré en Mot 6 sur la zone de sortie.
(Kx) transféré en Mot 5.
(1579) = Okkk, en Mot 3 de la zone de sortie.
(1577) = Adresse A, en Mot 1 de la zone de sortie.

1737 60 8002 1197 (8003) = adresse I
1197 44 1199 1750 (24)

3.32 Cas où l'adresse I est blanche.

1750 69 1553 1239 (8001) = (1553)

1239
C.C.S.
.... 1755

1553 69 1610 1763

1763 90 1717 1468 (25)

Carte suivante non-PASO:

1468 44 1622 1772

1772 69 1430 1787

1787 90 1717 1342 (25)

1342 60 1245 1149 (8003) = (1245)

1149
C.S.I.
.... 1504

1245 21 1951 1717 (25)

Carte suivante PASO:

(25) 1717 21 1578 1738

1738 69 1303 1561
1561 24 1031 1788

1788 60 1141 1295

(8003) = (1141) (17)

Si A est non-blanc alors que I l'est, il y a erreur dans l'enchaînement, d'où l'arrêt :
(1622) = 01 0320 1493

(8001) = 88.8 si la carte consultée est PASO
= 99.9 si la carte consultée est non-PASO.

(8003) = A₀
(1755) = (1553)
(8001) = 88.8 si la carte consultée est PASO
= 99.9 si la carte consultée est non-PASO.

Si A est non-blanc alors que I l'est, il y a erreur dans l'enchaînement, d'où l'arrêt :
(1622) = 01 0320 1493

(8001) = 88.8 si la carte consultée a un code K
= 99.9 si la carte consultée a un code non-K.

(8003) = AB99V
(1504) = (1245)
(1951) = AB99V en remplacement de A qui est blanc.
(1578) = AB99V ou bien I suivant la nature de la carte suivante.
(1031) = 999999999 +, pour permettre la poursuite de la traduction de la carte consultée.

1295
P.F.O.
.... 1592

Numérotation et perforation de la traduction de la carte K :

(A) EDI () Okkk KFx I

(1592) = (1141). En raison de la consultation précédente, la suite des instructions est prise en 1141 pour la poursuite de la traduction de la carte consultée.

3.33 Cas où l'adresse I est non-blanche.

(24) 1199 21 1578 1339 (1578) = adresse I.
1339 60 1334 1295 (8003) = (1334)

1295
P.F.O.
.... 1592

Numérotation et perforation de la traduction de la carte K :

(A) EDI (I) Okkk KFx I

(1592) = (1334). Puisqu'il n'y a pas eu consultation de la carte suivante, la suite est prise en 1334 pour lire la carte suivante.

3.4 Traduction d'une carte au code non-K

La traduction d'une carte non-K est faite d'une séquence de 3 cartes.

Si Mxx est un code non-ZZ, on aura :

(A) 00 (F) () (1430)=9999999999+
() RAD (A) () (1560)=9999999999+
() AAG (I) 17xx (1610)=9999999999+
Si Mxx est un code ZZ, on aura :
(A) RAD AB99T () (1430)=9999999999+
AB99T 00 Δ₁ Δ₂ (1560)=8888888888+
() AAG (I) 17xx (1610)=9999999999+

où 17xx est l'entrée sélectionnée lors de la recherche du code.

3.4.1 Partie commune à tous les codes non-K

Transferts sur REMARQUES.
(23) 1292 44 1345 1096

(8003) = adresse facteur F
Si F est blanc, donc 8003 nul, puisque le code est un macro-code non-K, il y a erreur d'écriture, d'où l'arrêt :
(1096) = 01 0330 1751

1345 20 1249 1602 (1249) = Adresse X
1602 30 0008 1574 (8003) = contenu de la colonne 51
1574 44 1439 1489 Si l'adresse F est symbolique, donc la colonne 51 perforée, le code doit être en V.
(26)
1439 69 1709 1512 (8001) = 88.8 si code en V
= 99.9 dans le cas contraire
1512 90 1489 1096 Si le code est non-V, il y a erreur d'écriture, d'où l'arrêt :
(26)
(1096) = 01 0330 1751

(26) 1489 30 0002 1395 (8003) = Adresse F
1395 60 8002 1603 (8002) = Adresse I.
1603 15 1249 1653
1653 21 1632 1539 Transfert de l'adresse facteur F en Mot 6 des différentes zones de sortie susceptibles d'être utilisées pour la perforation de la séquence de traduction.
1539 24 1682 1589
1589 24 1732 1639
1639 24 1782 1689
1689 24 1832 1739
1739 69 1954 1611 (8001) = (COP)
1611 24 1631 1789 Transfert du code-opération en Mot 5 des différentes zones de sortie susceptibles d'être utilisées pour la perforation de la séquence de traduction.
1789 24 1681 1240
1240 24 1731 1290
1290 24 1781 1340
1340 24 1831 1440

Transferts préliminaires.
Séparation des codes non-ZZ des codes en ZZ.

1440 60 1951 1661 (8003) = adresse A ou AB99V, cette constante ayant été construite après consultation de la carte actuelle lors de la traduction de la carte précédente.
1661 44 1465 1266
(27)
1266 60 1465 1149 (8003) = (1465)
(8003) = AB99U en remplacement de A qui était blanc.
(1504) = (1465) (27)
A ou son remplaçant en Mot 1 de la première carte de la séquence de traduction pour un code non-ZZ.

1149
C.S.I.
.... 1504

(27) 1465 21 1627 1490

1490 21 1777 1540

1540 24 1678 1590

1590 69 1756 1711

1711 24 1729 1640

1640 69 1560 1264

1264 90 1518 1319

(28)

3.42 Traduction pour un code non-ZZ

Première carte:

1319 69 1952 1662

1662 24 1628 1591

1591 60 1344 1399

1399
P.F.O.
.... 1592

Seconde carte:

1344 60 1297 1501

1501
P.F.O.
.... 1592

1297 60 1953 1712
1712 44 1515 1316
(29)

1515 21 1728 1641

1641 60 1334 1445

A ou son remplaçant en Mot 1 de la première carte de la séquence de traduction pour un code ZZ.

Egalement en Mot 2 de la seconde carte de la séquence de traduction dans le cas d'un code non-ZZ.

(1729) = 1750 ou Okkk, en Mot 3 de la troisième carte de la séquence de traduction dans les 2 cas : code ZZ ou non.

(8001) = 88.8 pour un code ZZ

= 99.9 pour un code non-ZZ

(1628) = adresse P. Cette adresse est non-blanche, le test 23 ayant permis d'écarter cette éventualité.

(8003) = (1344)

Numérotation et perforation de la première carte :

(A) 00 (F) () Mxx (P)
(1592) = (1344)

(8003) = (1297)

Numérotation et perforation de la seconde carte :

() RAD (A) () Mxx (P)
(1592) = (1297)

(8003) = adresse I

I est transféré en Mot 2

(8003) = (1334)

1445
P.F.O.
.... 1592

(29) 1316 69 1519 1239

1239
C.C.S.
.... 1755

1519 69 1610 1464

1464 90 1451 1569

(30)

1569 44 1622 1469
(31)

3.43 Traduction pour un code ZZ

Première carte:

(28) 1518 60 1624 1149

1149
C.S.I.
.... 1755

1624 21 1778 1690

1690 21 1827 1740

1740 60 1343 1197

1197
P.F.O.
.... 1592

Numérotation et perforation de la troisième carte :

() AAG (I) 17xx Mxx (P)
(1592) = (1334) (38)

Les 3 cartes de la séquence de traduction sont perforées et comme il n'y a pas eu consultation de la carte suivante, la suite est prise en 1334 pour lire la carte suivante.

(8001) = (1519)

(8003) = A_s

(1755) = (1519)

(8001) = 88.8 si le code de la carte consultée est PASO
= 99.9 dans le cas contraire.

Si A_s est non-blanc alors que I l'est, il y a erreur d'enchaînement, d'où l'arrêt :

(1622) = 01 0320 1490

(8003) = (1624)

(8003) = AB99T

(1755) = (1624)

AB99T en Mot 2 de la première carte, et

en Mot 1 de la seconde carte.

(8003) = (1343)

Numérotation et perforation de la première carte :

(A) RAD AB99T () MZZ A₁
(1592) = (1343)

Seconde carte

1345 60 1451 1751
1361 17 1753 1754
1362 24 1753 1750
1363 20 1753 1751
1364 6 1753 1753

1365 17 1753 1754
1366 20 1753 1751
1367 6 1753 1753

Troisième carte

1294 69 1247 1239

1239 17 1753 1754
1240 20 1753 1751
1241 6 1753 1753

1247 44 1451 1652
(30)
1252 69 1610 1314

1314 90 1451 1469
(30) (31)
1362 69 1430 1441

1441 90 1451 1146
(30)
1146 60 1349 1149

1149 17 1753 1754
1150 20 1753 1751
1151 6 1753 1753

1349 21 1951 1451 (30)

(30) 1451 21 1728 1491

1491 69 1303 1612
1612 24 1031 1541

(8003) = A_1
(8002) = A_2
 A_1 chargé en Mot 2 de la seconde
carte et,
 A_2 en Mot 3.
(8003) = (1294)

Numérotation et perforation de la
seconde carte de la séquence :
(1294) = (1294)

(9001) = (1247)

(8003) = A_3
(1755) = (1247)

(8001) = 88.8 si le code de la
carte consultée est PASO
= 99.9 dans le cas con-
traire.

(8001) = 88.8 si le code de la
carte consultée est K
= 99.9 s'il est non-K

(8003) = (1349)

(8003) = AB99S
(1755) = (1349)

(1951) = AB99S à la place de A_3
qui était blanc.
AB99S en Mot 2 de la troisième
carte.

(1031) = 999999999 + pour per-
mettre de poursuivre la traduc-

tion de la carte consultée.
(8003) = (1141)

1541 60 1141 1445

1445 17 1541 1445
P.F.O.
1542 20 1541 1445

Numérotation et perforation de la
troisième carte de la séquence :

(1542) = (1441) (10)
Les 3 cartes de la séquence de
traduction ayant été perforées,
la carte est prise en 1441 pour
poursuivre la traduction de la
carte précédemment consultée.

§ 4 TRADUCTION DE LA PSEUDO-
INSTRUCTION Z.

4.1 Transcodage du nombre Omnp de constantes
symboliques internes.

(32) 1091 60 1450 1665 (8003) = 00 0000 Omnp
1665 30 0001 1775 (1642) = 00 0000 0000
1775 20 1642 1595 (8003) = 00 0000 0000
1595 60 8003 1715 (1692) = 00 0000 0000
1715 30 0001 1576 (8003) = 00 0000 0000
1576 20 1692 1645 (1742) = 00 0000 0000
1645 60 8003 1765 (1692) = 00 0000 0000
1765 35 0004 1626 (8003) = 00 0000 0000
1626 21 1742 1695
1695 65 1692 1397
1397 35 0003 1466
1466 21 1692 1745
1745 65 1642 1447
1447 35 0001 1516
1516 10 1692 1497
1497 10 1742 1547 (8003) = 00 0000 Omnp
1547 10 1701 1566 (8003) = 00 909m 9H9p
1566 21 1929 1792 (1929) = (Omnp)^t en Mot 3 de la
zone de sortie.

4.2 Chargement de ANTE(S) en 1932

1792 65 1767 1676 (8002) = 00 0000 0001
1676 16 1450 1616
1616 46 1570 1620
1570 69 1726 1393 (8002) = 1 - mmp
(8001) = 6175836582 = (ANTE)_t
1620 69 1776 1393 (8001) = 6175836500 = (ANTE)_t
1393 24 1932 1443 (1932) = ANTE(S)
1443 60 1300 1601 (8003) = (1300)

Numérotation et perforation de la carte :

PDR Omnp CONSTANTE(S)

(1592) = (1300)

La compilation étant terminée, la suite est prise en 1300 pour la réinitialisation du Compileur.

§ 5 CORRECTIONS DE CERTAINES ERREURS.

5.1 Premier cas 01 0320 1493

L'adresse A de la carte suivante n'est pas blanche alors que l'adresse I de la carte présente l'est.

Dans la liste initiale, on reprend pour les corriger les 2 instructions dont l'enchaînement est défectueux. Dans la liste compilée, on retire les 2 dernières cartes perforées, c'est-à-dire le début de la séquence de traduction de la macro-instruction dont l'adresse I n'enchaîne pas avec la carte suivante. Pour la reprise du compilage on remplace les 2 cartes corrigées en tête et on appuie sur DEF.PROG.

La première instruction est alors prise en 1493.

1493 65 1012 1670 (8002) = 00 0000 00W* de la carte
1670 16 1525 1543
1543 20 1012 1751 (33) (1012) = W* de la carte, diminué de 2 unités en raison de 2 cartes de la liste compilée qui ont été écartées.

5.2 Second cas 01 0330 1751

L'adresse facteur F est blanche ou bien symbolique (colonne 51 perforée) alors que le macro-code n'est pas en W. Corriger la

carte erronée et la placer en tête de la liste à compiler avant d'appuyer sur DEF.PROG. La première instruction est alors prise en 1751.

(33) 1751 65 1450 1666 (8002) = 00 0000 0mmp
1666 16 1767 1593
1593 20 1450 1716 (1450) = 00 0000 0(mmp-1)
1716 60 1112 1720 (8003) = (AB99X)_t

1720 30 0001 1643 (8002) = x₂ 0 0000 0000

1643 65 8002 1766
1766 16 1770 1693
1693 45 1196 1597 (8002) = (x₂-9) 0 0000 0000

(36) 1196 65 1767 1743 (8002) = 0000000001

(37) 1743 15 1112 1793 (8002) = (AB99X)_t + 0000000001

1793 20 1112 1494 (1112) = (AB99Y)_t

1494 69 1303 1544

1544 24 1031 1334 (38) (1031) = 9999999999 +, pour permettre la traduction de la carte à lire.

En 1334 on reprend la lecture de la liste à compiler, donc de la carte corrigée.

(34) 1597 60 1112 1594 (8003) = (AB99X)_t

1594 11 1647 1644

1644 44 1697 1248 (8003) = (AB99X)_t - (03999)_t

(35) 1697 10 1694 1649

1649 44 1744 1196 (8003) = (AB99X)_t - (02999)_t

(36) 1744 65 1525 1743 (37) (8002) = 00 0000 0002

(35) 1248 66 1575 1743 (37) (8002) = 00 0100 0038 -

§6 SOUS-PROGRAMMES UTILISES

6.1 Sous-Programme C.C.S.

Si dans une carte de la liste à compiler, l'adresse instruction suivante I est blanche, un problème d'enchaînement qui n'existait pas pour la liste initiale, peut se poser dans la liste traduite du fait de la traduction par une séquence de trois

cartes le certaines macro-instructions.

Ce remplacement du blanc par une constante symbolique interne n'est nécessaire que lorsque la carte suivante sera traduite par une séquence de 3 cartes. Il est donc parfois nécessaire en cours de traduction d'une carte de consulter la carte suivante dans la liste initiale.

On utilise alors le Sous-Programme de Consultation de la Carte Suivante (O.C.S.).

A l'entrée, 8001 contient l'instruction de sortie du sous-programme.

1239 24 1755 1710 (1755) = Instruction de sortie du sous-programme.

1710 69 1721 1524 (38) (1031) = 8888888888 + signifie que la carte lue ne l'est que pour consultation, jusqu'à ce que son code opération ait été identifié.

1524 24 1031 1334

Après lecture de la carte et identification du code opération on charge de 8 ou 9 les différentes mémoires d'orientation suivant la nature du code :

PA50 ou non-PA50,
K ou non-K,
V ou non-V et
ZZ ou non-ZZ.

(8001) = 8888888888 +

(38)	1334	70	1973	1862
....	..			
....	..			
1553	69	1631	1336	
1336	68	1140	1141	
		(16)	(17)	

1140 60 1951 1755 (8003) = A, adresse A de la carte suivante.

1755 Instruction de sortie du sous-programme.

S. Sous-programme O.C.S.I.

Le but de ce sous-programme est de construire les Constantes Symboliques Internes nécessaires à l'enchaînement des séquences de traduction. Le nombre de ces constantes doit rester inférieur à 400.

Elles sont de la forme : AB99X avec :

- A = 0 ou 1
- B = 2,3,4,5,6,7,8 ou 9
- X = 0,2,3,9, Z, Y, X, W, V, U, T, S, R, Q, P, O, N, M, L, K, I, H, G, F, E, D, C ou B.

A partir de la forme initiale 02999, X est formé par soustraction et A et B par addition.

A l'entrée, 8003 contient l'instruction de sortie du sous-programme.

6.21 Contrôle du nombre des constantes.

1149 21 1504 1762 (1504) = Instruction de sortie du sous-programme.

1762 65 1450 1514 (8002) = 00 0000 0000, nombre des constantes utilisées.

1514 15 1767 1674 (8002) = (mnp + 1) - 400

1674 16 1691 1495 Si la nouvelle constante que l'on désire construire a un rang supérieur à 399, il y a arrêt :

1495 45 1098 1449 (1449) = 01 0311 8000.

1098 15 8001 1564 (1450) = 00 0000 0(mnp+1)

1564 20 1450 1703

6.22 Construction de la nouvelle constante.

1703 60 1112 1568 (8003) = (AB99X)[†] - 00000000001

1568 11 1767 1724 (1112) = (8003) = (AB99X)[†]

1724 21 1112 1565

6.23 Tests sur la nouvelle constante.

1565 30 0001 1774 (8002) = x₁0 0000 0000

1774 65 8002 1741 (8002) = (x₁-1)0 0000 0000

1741 16 1394 1499 (8003) = (AB99X)[†] avec x₁≠1, donc valable.

1499 45 1702 1753

1702 60 1112 1504

1504 Instruction de Sortie du Sous-programme O.S.I.

(40) 1753 60 1112 1618 x₁=1, il faut vérifier que x₂ est supérieur à 61.

1618 30 0002 1475 (8002) = x₂x₁ 0000 0000

1475 65 8002 1791

1791 16 1444 1549 (8002) = (x₂x₁-61) 0000 0000

1549 45 1752 1554 (41)

1752 60 1112 1668

1668 11 1525 1392 (8003) = (AB99X)[†] - 00000000002

(43) 1392 21 1112 1504 (1112) = (8003) = (AB99X)_t
 1504 Instruction
 de sortie du sous-
 programme C.S.I.

(41) 1554 65 1112 1718
 1718 35 0004 1442
 1442 65 8003 1599
 1599 16 1604 1614
 1614 45 1768 1619
 (42)
 1768 60 1112 1669
 1669 10 1575 1392 (43)
 (43) 1619 60 1625 1392 (43) (8003) = (13000)_t

(8002) = 71 1.²₈ 1592
 (1719) = 20 1.³₈ 1615
 (1592) = Instruction de sortie
 du sous-programme.
 (8002) = 00 0000 00M^o = numéro
 de la carte précédemment perforée.
 (8002) = 00 0000 00(M^o+1)
 Transfert du nouveau numéro d'
 ordre en Mot 9 de la zone qui
 doit être perforée.
 (1012) = 00 0000 00(M^o+1)
 Perforation de la zone choisie.

1764 69 1719 1675
 1675 22 1719 1725
 1725 21 1592 1545
 1545 65 1012 1769
 1769 15 1767 1719
 1719 20 1.³₈ 1615
 1615 24 1012 1651
 1651 71 1.²₇ 1592
 1592 Instruction
 de sortie du sous-
 programme P.F.O..

6.3 Sous-Programme P.F.O.

A chaque modèle de carte perforée correspond une zone de perforation (voir schéma au §). Les différents Mots de l'une de ces dix zones ayant été chargés des quantités convenable, le but du sous-programme P.F.O. est :

- 1) de numéroté la carte en Mot 9
 - 2) de perforer cette zone sur la carte de traduction.
- A chacune de ces dix zones correspond une entrée particulière dans le sous-programme P.F.O.. De plus, à l'entrée, 8003 contient l'Instruction de sortie du sous-programme.

1039 15 1492 1347 (8002) = 71 1527 1592
 1295 15 1148 1347 (8002) = 71 1577 1592
 1399 15 1654 1347 (8002) = 71 1627 1592
 1501 15 1704 1347 (8002) = 71 1677 1592
 1445 15 1198 1347 (8002) = 71 1727 1592
 1197 15 1551 1347 (8002) = 71 1777 1592
 1299 15 1754 1347 (8002) = 71 1827 1592
 1723 15 1526 1347 (8002) = 71 1877 1592
 1601 15 1664 1347 (8002) = 71 1927 1592
 1289 15 1542 1347 (8002) = 71 1977 1592
 1347 20 1651 1714 (1651) = 71 1.²₇ 1592
 1714 15 1508 1764

BIBLIOGRAPHIE

Programme I.B.M. DOPSR 02 1003

Multiple Précision Division de P. RABINOWITZ
A.C.M. volume 4/Number 2/ February 1961

Introduction à l'algèbre supérieure et au calcul numérique algébrique de L. DERWIDUE

Thèse de H. OULES
Etude des outillages de programmation de C A T - 1

Thèse de S. Aoustin
Etude de méthodes numériques directes pour la résolution de systèmes d'équations linéaires
sur I.B.M. 650 standard

Thèse de H. BAHUREL
Etude d'un langage symbolique à macro-instructions et analyse pour I BM 650 standard

TABLE DES MATIERES

	PAGES
INTRODUCTION	1
LISTE DES PRINCIPALES NOTATIONS & ABBREVIATIONS	3
CHAPITRE I - NOMBRES - VIRGULE FLOTTANTE MULTIPLE PRECISION - SCHEMA D'UNE ARITHMETIQUE POUR 650 -	
SECTION 1.1 - REPRESENTATION DES NOMBRES VFMP -	
OPERATIONS ARITHMETIQUES	9
1.11 - Représentation des nombres VFMP	9
1.111 - Introduction	9
1.112 - Conditions d'utilisation	9
1.113 - Représentation de l'exposant - Remarques	10
1.114 - Représentation de la mantisse	11
- 1 - Explication de M	11
- 2 - Mantisse normalisée	11
- 3 - Représentation de M	12
- 4 - Décalages à droite	12
1.115 - En résumé	13
1.116 - Remarques	13
1.12 - Opérations sur nombres VFMP	13
1.121 - Addition	14
- 1 - Représentation de l'addition - Règles d'addition	14
- 2 - Remarques	14
- 3 - Opérations se déduisant de l'addition	15
1.122 - Multiplication	15
- 1 - Expression de l'exposant du produit	15
- 2 - Expression de la mantisse du produit	15
1.123 - Division	18
- 1 - Expression de l'exposant du quotient	18
- 2 - Calcul de la mantisse par utilisation de l'algorithme manuel	18
- 3 - Calcul de la mantisse par développement en série	21
- 4 - Comparaison des 2 méthodes	22
SECTION 1.2 - REALISATION D'UNE ARITHMETIQUE REELLE	23
1.21 - Disposition des nombres en mémoire	23
1.22 - Pseudo-Bloc de Calcul	25
1.221 - Schéma d'une opération sur 650	25
1.222 - Schéma d'une opération dans l'Arithmétique	25
1.223 - Pseudo-Accumulateur	25
1.224 - Pseudo-Distributeur	26
1.225 - Sous-Programme d'Opérations	26
1.23 - Schéma de l'Arithmétique	27
1.231 - Présentation des données à l'entrée	27
1.232 - Présentation des résultats à la sortie	27
1.233 - Sous-Programme d'Entrée	27
1.234 - Sous-Programme de Transfert	28
- 1 - Vers le Pseudo-Accumulateur	28
- 2 - Vers la mémoire générale	29
1.235 - Sous-Programme d'Opérations	29
- 1 - Addition - Règles d'addition	30
Annexe A	31
Annexe B	33
- 2 - Multiplication	35
- 3 - Division - Chargement du second opérande	36
Cas particuliers	37
Calcul de l'exposant	37
Calcul de la mantisse par utilisation de l'algorithme manuelle	37
Calcul de la mantisse par développement en série	38
Comparaison de ces deux méthodes	40
1.236 - Sous-Programme de sortie	41
- 1 - Normalisation du résultat	41
- 2 - Mise en place de l'exposant	42

SECTION 1.3 - REALISATION D'UNE ARITHMETIQUE COMPLEXE	43
1.31 - Structure des nombres complexes VFMP	43
1.32 - Disposition des nombres complexes en mémoire	43
1.33 - Pseudo Bloc de Calcul	45
1.331 - Schéma d'une opération	45
1.332 - Pseudo-Accumulateur Complexe	45
1.333 - Pseudo-Distributeur Complexe	46
1.34 - Schéma de l'Arithmétique	46
1.341 - Présentation des données à l'entrée	46
1.342 - Présentation des résultats à la sortie	46
1.343 - Sous-Programme d'Entrée	47
1.344 - Sous-Programme de Transfert	47
- 1 - Vers le Pseudo-Accumulateur Complexe	47
- 2 - Vers la mémoire générale	47
1.345 - Sous-Programmes d'Opération	48
- 1 - Addition	48
- 2 - Soustraction	48
- 3 - Multiplication	48
- 4 - Division	49

CHAPITRE II - ETUDE DES ARITHMETIQUES REALISEES EXEMPLES D'UTILISATION -

SECTION 2.1 - ARITHMETIQUE POUR NOMBRES REELS	55
2.11 - Arithmétique Double Précision (A.D.P.)	55
2.111 - Nombres traités	55
- 1 - Structure	55
- 2 - Disposition en mémoire	56
2.112 - Pseudo-Bloc de Calcul	56
2.113 - Etude de l'Arithmétique	56
- 1 - Encombrement	56
- 2 - Présentation	56
- 3 - Utilisation	56
2.114 - Temps de Calcul	58
2.12 - Arithmétique mixte Simple-Double Précision (A.S.D.P.)	59
2.121 - Nombres traités	59
- 1 - Structure	59
- 2 - Disposition en mémoire	59
2.122 - Pseudo-Bloc de Calcul	60
- 1 - En Double Précision	60
- 2 - En Simple Précision	60
2.123 - Etude de l'Arithmétique	60
- 1 - Encombrement	60
- 2 - Présentation	60
- 3 - Utilisation en double précision	61
- 4 - Utilisation en simple précision	61
2.124 - Temps de calcul	62
2.13 - Arithmétique Triple Précision (A.T.P.)	63
2.131 - Nombres traités	63
- 1 - Structure	63
- 2 - Disposition en mémoire	64
2.132 - Pseudo-Bloc de Calcul	64
2.133 - Etude de l'Arithmétique	64
- 1 - Encombrement	64
- 2 - Présentation	64
- 3 - Utilisation	65
2.134 - Temps de calcul	66
SECTION 2.2 - ARITHMETIQUES POUR NOMBRES COMPLEXES	69
2.21 - Arithmétique Complexe Simple Précision (A.Cx.S.P.)	69
2.211 - Nombres traités	69
- 1 - Structure	69
- 2 - Disposition en mémoire	69
2.212 - Pseudo-Bloc de Calcul	69
2.213 - Etude de l'Arithmétique	70
- 1 - Encombrement	70
- 2 - Présentation	70
- 3 - Utilisation	70
2.214 - Temps de calcul	71
2.22 - Arithmétique Complexe Double Précision (A.Cx.D.P.)	72
2.221 - Nombres traités	72
- 1 - Structure	72
- 2 - Disposition en mémoire	73
2.222 - Pseudo-Bloc de Calcul	73
2.223 - Etude de l'Arithmétique	73

	PAGES
- 1 - Encombrement	73
- 2 - Présentation	74
- 3 - Utilisation	74
2.224 - Temps de calcul	75
2.23 - Arithmétique Complexe Mixte Simple - Double Précision (A.Cx.S.D.P.)	76
2.231 - Nombres traités	76
- 1 - Structure en Simple Précision	76
- 2 - Structure en Double Précision	76
- 3 - Disposition en mémoire	76
2.232 - Pseudo-Bloc de Calcul	77
- 1 - En Simple Précision	77
- 2 - En Double Précision	77
2.233 - Etude de l'Arithmétique	78
- 1 - Encombrement	78
- 2 - Présentation	78
- 3 - Utilisation en double précision	78
- 4 - Utilisation en simple précision	78
2.234 - Temps de calcul	79
SECTION 2.3 - EXEMPLES D'UTILISATION -	81
2.31 - Table de racine carrée en double précision	81
2.311 - Méthode utilisée.....	81
2.312 - Organigramme	82
2.313 - Utilisation du programme	81
- 1 - Caractéristiques du programme	81
- 2 - Format des cartes données	81
- 3 - Mode d'emploi	83
2.314 - Résultats	83
2.315 - Analyse du programme	85
- 1 - Mémoires à contenu fixe	85
- 2 - Mémoires à contenu variable	85
- 3 - Analyse des instructions	86
2.32 - Calcul de $e^{(x + iy)}$ en double précision	89
2.321 - Méthode utilisée	89
2.322 - Organigramme	90
2.323 - Utilisation du programme	89
- 1 - Caractéristiques du programme	89
- 2 - Format des cartes données	91
- 3 - Mode d'emploi	91
2.324 - Résultats	91
2.325 - Analyse du programme	94
- 1 - Mémoires à contenu fixe	
- 2 - Mémoires à contenu variable	
- 3 - Analyse des instructions	
CHAPITRE III - REGLES DE PROGRAMMATION POUR UTILISATION DES MACRO-CODES ET DU COMPILEUR MULTIPLE PRECISION -	
SECTION 3.1 - REGLES PARTICULIERES DE PROGRAMMATION	101
3.11 - Règles générales	101
3.111 - Codes opération	101
3.112 - Liste des codes opérations	101
3.113 - Adresses	103
3.114 - Autres considérations	104
3.115 - Remarques	104
3.12 - Règles particulières	104
3.121 - Cas général des macro-instructions arithmétiques	104
3.122 - Exceptions au cas général	105
- 1 - Codes en V	105
- 2 - Codes en Z Z	105
3.123 - Macro-instructions fonctionnelles	106
3.124 - Pseudo-instructions	107
- 1 - Instructions standard PASO	107
- 2 - BBL de 1ère et 2ème espèces	107
- 3 - Pseudo-instruction Z	108
3.125 - Remarques générales	108
SECTION 3.2 - BIBLIOTHEQUES	109
3.21 - Bibliothèques Multiple Précision	109
3.211 - Nomenclature des bibliothèques	109
3.212 - Plan d'encombrement des bibliothèques	109
3.22 - Choix de la bibliothèque	110
3.221 - Premier cas	110
3.222 - Deuxième cas	110

	PAGES
3.223 - Troisième cas	111
3.224 - Quatrième cas	111
3.225 - Remarques	112
3.23 - Utilisation des sous-programmes de bibliothèque	112
3.231 - Arithmétiques Réelles	112
- 1 - A.D.P.	112
- 2 - A.S.D.P.	113
- 3 - A.T.P.	114
3.232 - Arithmétiques Complexes	115
- 1 - A.Cx.S.P.	115
- 2 - A.Cx.D.P.	115
- 3 - A.Cx.S.D.P.	116
3.233 - Fonctions double précision	117
- 1 - Argument	117
- 2 - Programmation d'une opération fonctionnelle	118
SECTION 3.3 - GUIDE OPERATEUR	121
3.31 - Premier Passage : Compilation	121
3.311 - Pratique de la Compilation	121
- 1 - Préparation de la liste	121
- 2 - Unité 533	121
- 3 - Pupitre du 650	122
- 4 - Mise en marche	122
- 5 - Exécution de la compilation	122
- 6 - Fin de la compilation	123
3.312 - Résultats de la Compilation	124
- 1 - Généralités	124
- 2 - Remarques générales	124
- 3 - Colonnes REMARQUES	124
3.313 - Traduction des macro-instructions	124
- 1 - Macro-instructions arithmétiques	124
- 2 - Macro-instructions en Z Z	126
- 3 - Instructions fonctionnelles	127
- 4 - Pseudo-instruction BBL de première espèce	127
- 5 - Pseudo-instruction BBL de seconde espèce	128
- 6 - Pseudo-instruction Z	128
3.32 - Second passage : Assemblage	128
3.33 - Utilisation du programme assemblé	129
3.331 - Chargement du tambour	129
3.332 - Arrêts programmés des bibliothèques	129
SECTION 3.4 - EXEMPLES	131
3.41 - Table de racine carrée en double précision	131
3.411 - Ecriture du programme	131
3.412 - Résultats de la compilation	134
3.413 - Utilisation du programme et résultats	134
3.42 - Calcul de $e^{(x + iy)}$ en double précision	139
3.421 - Ecriture du programme	139
3.422 - Résultats de la compilation	145
3.423 - Utilisation du programme et résultats	146
3.43 - Calcul direct de $e^x \cos y$ et $e^x \sin y$ en double précision	151
3.431 - Ecriture du programme	151
3.432 - Résultats de la compilation	153
3.433 - Utilisation du programme	154
3.434 - Résultats	154

CHAPITRE IV - COMPILEUR POUR CODES MULTIPLE PRECISION -

SECTION 4.1 - TABLEAU SYNOPTIQUE DE LA COMPILATION D'UN PROGRAMME	159
4.11 - Organigramme général du Compileur	159
4.12 - Tableau synoptique de la compilation	160
SECTION 4.2 - REMARQUES CONCERNANT LE COMPILEUR	165
4.21 - Tables des codes opérations	165
4.22 - Mémoires intermédiaires d'entrée et de sortie	166
4.221 - Mémoires d'entrée	166
4.222 - Mémoires de sortie	167
4.23 - Plan d'encombrement du compileur	169

CHAPITRE V - ANNEXES

1 - REMARQUES POUR L'UTILISATION DES ANNEXES	171
- Structure d'une ligne	171
- Représentation d'un Sous-Programme	172

2 - ANNEXES -

ANNEXE 1 - Arithmétique Double Précision	173
ANNEXE 2 - Arithmétique mixte Simple Double Précision	185
ANNEXE 3 - Arithmétique Triple Précision	197
ANNEXE 4 - Arithmétique Complexe Simple Précision	211
ANNEXE 5 - Arithmétique Complexe Double Précision	219
ANNEXE 6 - Arithmétique Complexe mixte Simple Double Précision	227
ANNEXE 7 - COMPILEUR	239

BIBLIOGRAPHIE -

Deuxième Thèse

Propositions données par la Faculté :

LOCALISATION DES VALEURS PROPRES
D'UNE MATRICE CARREE

VU :

Le Directeur de la Thèse
Président du Jury :

G. BRILLOUET

VU et APPROUVE

Le Doyen de la Faculté des Sciences :

G. BRILLOUET

VU et PERMIS d'IMPRIMER

Le Recteur de l'Académie de Nantes :

13 OCT. 1964

M. SCHMITT

